# **Chuẩn bị**

Để bắt đầu, cần chuẩn bị môi trường phát triển Laravel trên máy tính của mình. Hãy đảm bảo đã cài đặt các yêu cầu sau đây:

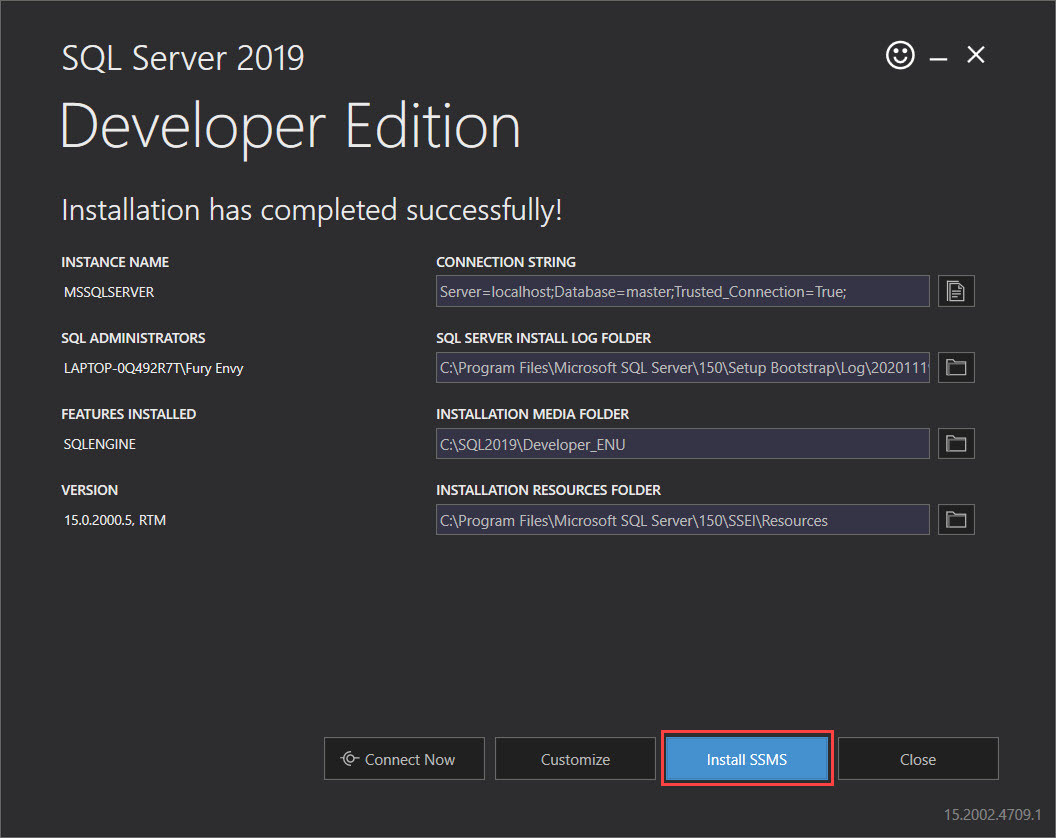
1. Phiên bản PHP tương thích với Laravel.
2. Composer - Trình quản lý gói PHP.

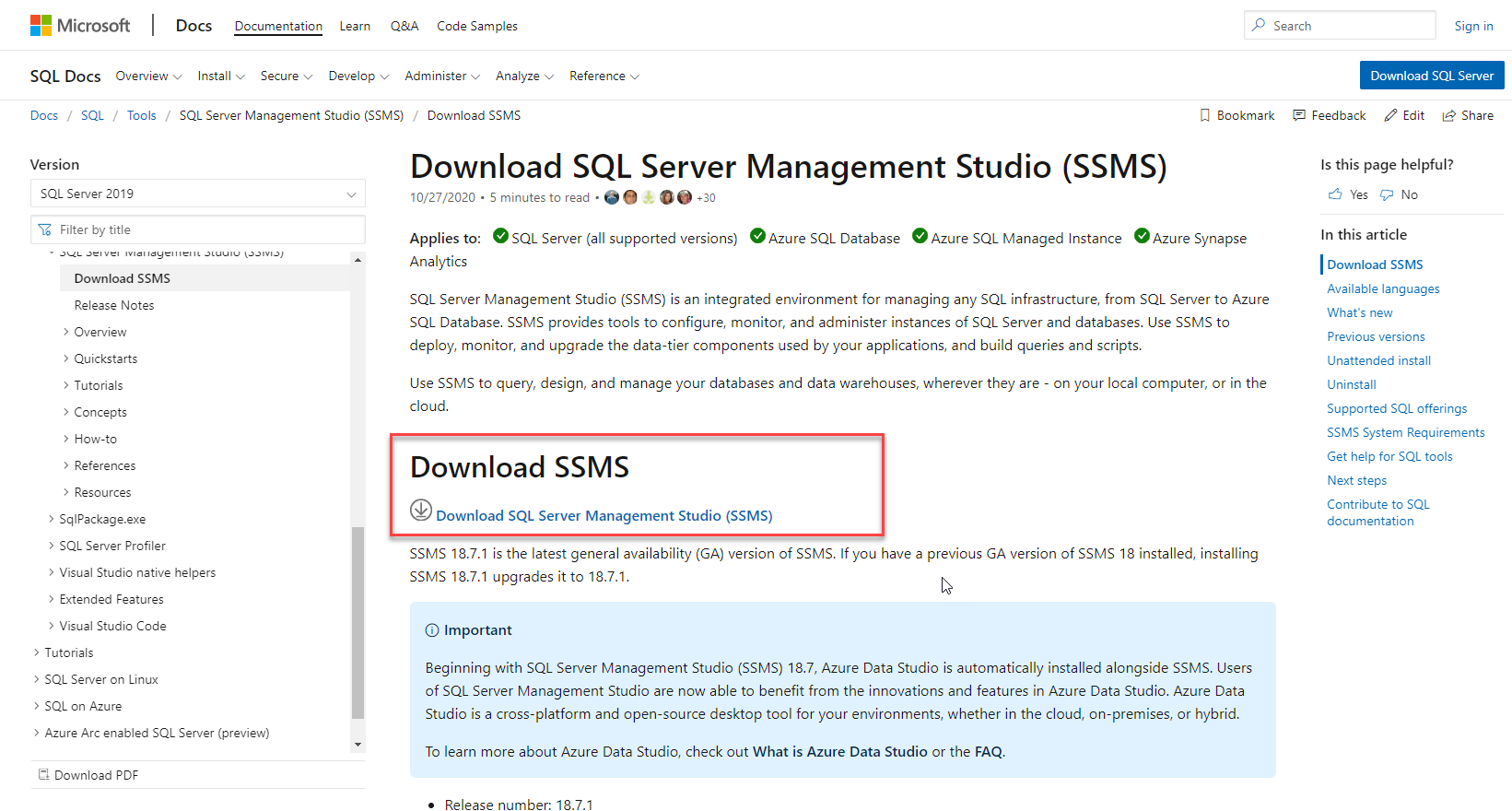
Xem cách cài đặt PHP8 và Composer tại đậy: [Click](https://www.youtube.com/watch?v=d4HsLO0BfJs&t=1329s)

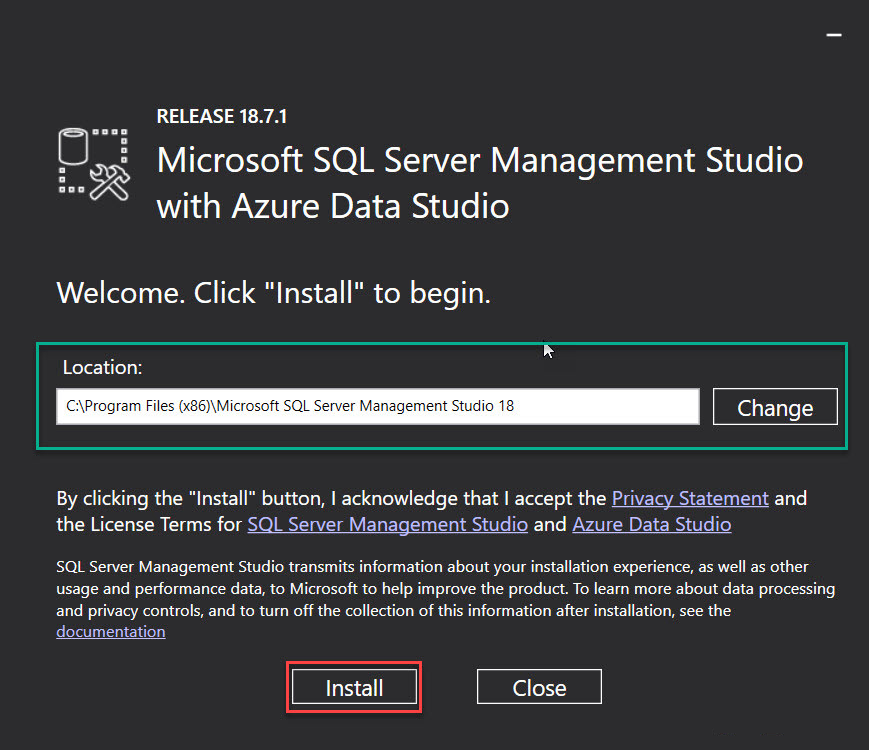
1. SQL Server Management Studio (SSMS)

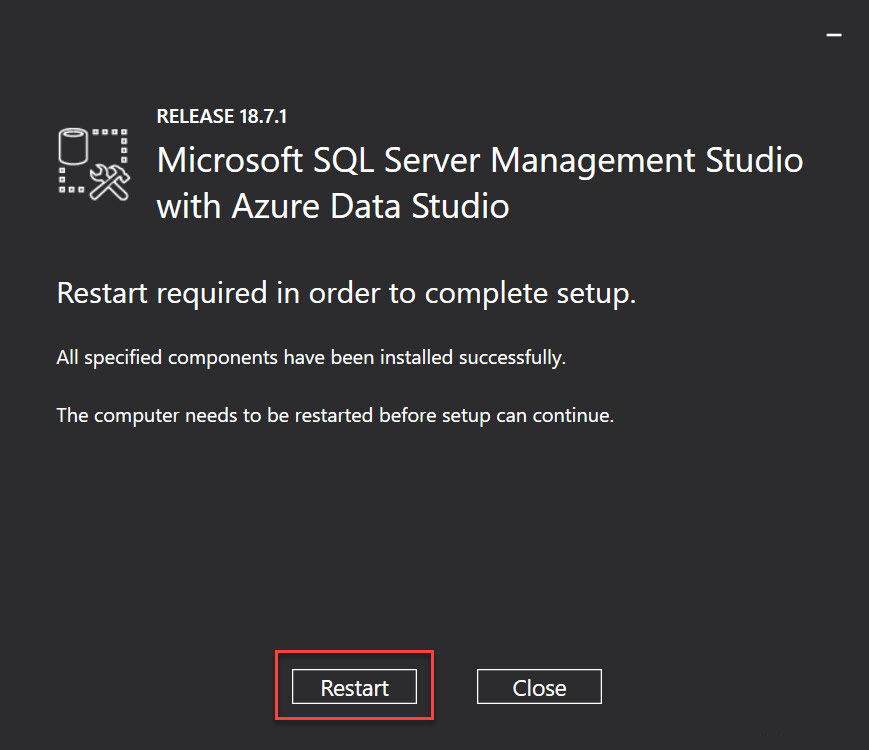
Tải SSMS tại đây: [Click](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15)

Hướng dẫn kết nối SqlServer với PHP - Laravel: [Click](https://viblo.asia/p/cach-ket-noi-laravel-vs-sql-server-fix-bug-cho-web-laravel-sau-khi-cai-lai-win-aWj53jyel6m)



Tải tập tin cài đặt SSMS như hình rồi khởi chạy

Chọn vị trí cài đăt > **Install**

**Restart** để hoàn thành việc cài đặt SSMS

# 

# **Tạo dự án Laravel**

| Bước 1 | Truy cập thực mục chứa project, mở cmd lên và type “laravel new [tên-project]”. Sau khi type, ấn enter và type tiếp là “none” |
| --- | --- |
| Bước 2 | Type “0” vì sử dụng PHPUnit |
| Bước 3 |  |
| Bước 4 | Type “sqlsrv” do dự này sẽ sử dụng Sql Server |
| Bước 5 | Dự án sẽ báo đã được tạo project thành công, hãy truy cập vào folder của project. |

# **Hướng dẫn**

## **Bước 1:** Tạo cơ sở dữ liệu và cấu hình .env

- Tạo một cơ sở dữ liệu trên máy chủ SQLServer của bạn. Sau đó, cập nhật file .env:

* DB\_DATABASE=tên\_cơ\_sở\_dữ\_liệu
* DB\_USERNAME=tên\_người\_dùng [Nếu sử dụng

## **Bước 2:** Tạo những models, controller và migrations cần thiết:

* Tạo migrations

php artisan make:model RawMaterialType -c

php artisan make:model RawMaterial -cr

php artisan make:model Customer-cr

php artisan make:model Order -cr

php artisan make:model ContentSimple -c

php artisan make:model ContentPack -c

php artisan make:model OrderMakeOrExpedition -c

* Trong lệnh php artisan make:model RawMaterial -cr, các tham số -mcr có ý nghĩa sau:
  + -c hoặc --controller: Thêm tham số này sẽ yêu cầu Laravel tạo một controller mới cho model. Controller này mặc định sẽ không có bất kỳ phương thức nào.
  + -r hoặc --resource: Khi sử dụng cùng với -c, tham số này sẽ yêu cầu Laravel tạo một controller tài nguyên (resource controller) với tất cả các phương thức cần thiết cho CRUD (tức là: index, create, store, show, edit, update, destroy).
* Vì vậy, lệnh php artisan make:model RawMaterial -mcr sẽ tạo ra:
  + Một model tên là RawMaterial .
  + Một resource controller cho model RawMaterial ,với tất cả các phương thức CRUD.

## **Bước 3:** Thêm dữ liệu

Để thêm dữ liệu vào Database, chúng ta sẽ mở SqlServer lên và vào database rồi chạy câu query

| insert into Customer (Id\_Customer, Name\_Customer, Name\_Contact, Email, Phone, Address, ZipCode, FK\_Id\_Mode\_Transport, Time\_Reception) values  (1, N'Nội bộ - Lưu kho (Lê Đình Tú)', N'Tus', 'ldt907@gmail.com', '0865176605', N'Đồng Nhân, Hai Bà Trưng, Hà Nội', '11608', '0', N'09/07/2023'),  (2, N'Trương Mỹ Hoa', N'Mic', 'tmh307@gmail.com', '0832498756', N'Lò Đúc, Hai Bà Trưng, Hà Nội', '11609', '1', N'03/07/2022'),  (3, N'Nguyễn Hải Tùng', N'Tũn', 'nht3107@gmail.com', '0938591792', N'Trần Khát Chân, Hai Bà Trưng, Hà Nội', '11623', '2', N'31/07/2023'),  (4, N'Hoàng Thanh Thủy', N'Tthuy', 'htt1412@gmail.com', '0393056789', N'Hương Viên, Hai Bà Trưng, Hà Nội', '11608', '3', N'14/12/2022'),  (5, N'Trần Công Thành', N'TTran', 'tct@gmail.com', '0912087956', N'Kim Ngưu, Hai Bà Trưng, Hà Nội', '11623', '3', N'04/08/2023'); |
| --- |

## **Bước 4:** Tạo HomeController.

php artisan make:controller HomeController

## **Bước 5:** Bước định nghĩa Routes.

| *use App\Http\Controllers\AuthController;*  *use App\Http\Controllers\ContentSimpleController;*  *use App\Http\Controllers\CustomerController;*  *use App\Http\Controllers\HomeController;*  *use App\Http\Controllers\OrderController;*  *use App\Http\Controllers\RawMaterialController;*  *use App\Http\Controllers\ContentPackController;*  *use App\Http\Controllers\ProductionStationLineController;*  *use App\Http\Controllers\StationController;*  *use App\Http\Controllers\WareController;*  *use App\Http\Controllers\OrderMakeOrExpeditionController;*  *use Illuminate\Support\Facades\Route;*  *Route::get('/login', [AuthController::class, 'login'])->name('login');*  *Route::post('/login', [AuthController::class, 'checklogin'])->name('checkLogin');*  *Route::get('/logout', [AuthController::class, 'logout'])->name('logout');*  *Route::middleware('checklogin')->group(function () {*  *Route::get('/', [HomeController::class, 'index'])->name('index');*  *Route::resource('rawMaterials', RawMaterialController::class);*  *Route::resource('orders', OrderController::class);*  *Route::resource('customers', CustomerController::class);*  *Route::resource('stations', StationController::class);*  *Route::resource('productionStationLines', ProductionStationLineController::class);*  *Route::group(['prefix' => 'rawMaterials', 'as' => 'rawMaterials.'], function () {*  *Route::post('/showMaterials', [RawMaterialController::class, 'showRawMaterialsByType'])->name('showMaterials');*  *Route::post('/showUnit', [RawMaterialController::class, 'showUnit'])->name('showUnit');*  *});*  *Route::group(['prefix' => 'packs', 'as' => 'packs.'], function () {*  *Route::get('/', [ContentPackController::class, 'index'])->name('index');*  *Route::get('/create', [ContentPackController::class, 'create'])->name('create');*  *Route::get('/createPack', [ContentPackController::class, 'createPack'])->name('createPack');*  *Route::post('/storePack', [ContentPackController::class, 'storePack'])->name('storePack');*  *Route::post('/storeDetail', [ContentPackController::class, 'storeDetailContentSimpleOfPack'])->name('storeDetailContentSimpleOfPack');*  *Route::post('/deletePack', [ContentPackController::class, 'deletePack'])->name('deletePack');*  *});*  *Route::group(['prefix' => 'orders', 'as' => 'orders.'], function () {*  *Route::post('/storeOrder', [OrderController::class, 'storeOrder'])->name('storeOrder');*  *Route::post('/updateOrder', [OrderController::class, 'updateOrder'])->name('updateOrder');*  *});*  *Route::group(['prefix' => 'simples', 'as' => 'simples.'], function () {*  *Route::post('/addSimple', [ContentSimpleController::class, 'addSimple'])->name('addSimple');*  *Route::post('/storeSimple', [ContentSimpleController::class, 'storeSimple'])->name('storeSimple');*  *Route::post('/deleteSimple', [ContentSimpleController::class, 'deleteSimple'])->name('deleteSimple');*  *Route::post('/updateSimple', [ContentSimpleController::class, 'updateSimple'])->name('updateSimple');*  *});*  *Route::group(['prefix' => 'productionStationLines', 'as' => 'productionStationLines'], function () {*  *Route::post('/{id}', [ProductionStationLineController::class, 'store'])->name('store');*  *Route::post('/update/{id}', [ProductionStationLineController::class, 'update'])->name('update');*  *});*  *Route::group(['prefix' => 'makesOrExpeditions', 'as' => 'makesOrExpeditions.'], function () {*  *Route::get('/', [OrderMakeOrExpeditionController::class, 'index'])->name('index');*  *Route::get('/create', [OrderMakeOrExpeditionController::class, 'create'])->name('create');*  *Route::post('/store', [OrderMakeOrExpeditionController::class, 'store'])->name('store');*  *Route::post('/show', [OrderMakeOrExpeditionController::class, 'show'])->name('show');*  *Route::post('/destroy', [OrderMakeOrExpeditionController::class, 'destroy'])->name('destroy');*  *});*  *Route::group(['prefix' => 'wares', 'as' => 'wares.'], function () {*  *Route::get('/', [WareController::class, 'index'])->name('index');*  *Route::get('/create', [WareController::class, 'create'])->name('create');*  *Route::get('/{id}', [WareController::class, 'show'])->name('show');*  *Route::post('createWare', [WareController::class, 'createWare']);*  *Route::post('show', [WareController::class, 'showDetails']);*  *Route::post('setCellStatus', [WareController::class, 'setCellStatus']);*  *});*  *Route::post('/stations/showStation', [StationController::class, 'showStationTypeById'])->name('showStationTypeById');*  *});* |
| --- |

**Trong file code trên:**

* “Route::get('/login', [AuthController::class, 'login'])->name('login');”
  + Định nghĩa một route POST với đường dẫn '/login' và liên kết nó với phương thức 'login' trong AuthController.
  + Route này được đặt tên là 'login' để có thể sử dụng tên này khi cần truy cập vào route.
* “Route::get('/logout', [AuthController::class, 'logout'])->name('logout');”
  + Định nghĩa một route POST với đường dẫn '/login' và liên kết nó với phương thức ‘checkLogin’ trong AuthController.
  + Route này được đặt tên là 'checkLogin' để có thể sử dụng tên này khi cần truy cập vào route.
* Route::get('/logout', [AuthController::class, 'logout'])->name('logout');
  + Định nghĩa một route GET với đường dẫn '/logout' và liên kết nó với phương thức 'logout' trong AuthController.
  + Route này được đặt tên là 'logout' để có thể sử dụng tên này khi cần truy cập vào route.
* “Route::middleware” : Sử dụng để phân quyền, người dùng bắt buộc phải đăng nhập tài khoản thì mới sử dụng chương trình được.
* Route::get('/', [HomeController::class, 'index'])->name('index');
  + Định nghĩa một route GET với đường dẫn '/' (gốc) và liên kết nó với phương thức 'index' trong HomeController.
  + Route này được đặt tên là 'index' để có thể sử dụng tên này khi cần truy cập vào route.
* “Route::resource('rawMaterials', RawMaterialController::class);” Sẽ gọi tất cả các hàm (index, create, store, edit, update, destroy, show) trong lớp RawMaterialController.
* “Route::resource('orders', OrderController::class);” Sẽ gọi tất cả các hàm (index, create, store, edit, update, destroy, show) trong lớp OrderController.
* “Route::resource('customers', CustomerController::class);” Sẽ gọi tất cả các hàm (index, create, store, edit, update, destroy, show) trong lớp CustomerController.
* “Route::resource(‘stations’, StationController::class);” Sẽ gọi tất cả các hàm (index, create, store, edit, update, destroy, show) trong lớp StationController.
* “Route::resource('productionStationLines', ProductionStationLineController::class);” Sẽ gọi tất cả các hàm (index, create, store, edit, update, destroy, show) trong lớp ProductionStationLineController.
* Route::group(['prefix' => 'rawMaterials', 'as' => 'rawMaterials.'], function () { ... });
  + Định nghĩa một nhóm route có tiền tố (prefix) là 'rawMaterials' và đặt tên cho nhóm route là 'rawMaterials'.
  + Các route bên trong nhóm này có đường dẫn sẽ có tiền tố 'rawMaterials' và tên route sẽ có tiền tố 'rawMaterials'.
* Route::group(['prefix' => 'packs', 'as' => 'packs.'], function () { ... });
  + Tương tự như Group Route trên, định nghĩa một nhóm route với tiền tố 'packs' và đặt tên nhóm là 'packs'.
* Route::group(['prefix' => 'orders', 'as' => 'orders.'], function () { ... });
  + Tương tự như Group Route trên, định nghĩa một nhóm route với tiền tố 'orders' và đặt tên nhóm là ‘orders’.
* Route::group(['prefix' => 'simples', 'as' => 'simples.'], function () { ... });
  + Tương tự như Group Route trên, định nghĩa một nhóm route với tiền tố 'simples' và đặt tên nhóm là 'simples'.
* Route::group(['prefix' => 'productionStationLines', 'as' => 'productionStationLines.'], function () { ... });
  + Tương tự như Group Route trên, định nghĩa một nhóm route với tiền tố ‘productionStationLines' và đặt tên nhóm là 'productionStationLines’.
* Route::group(['prefix' => 'makesOrExpeditions', 'as' => 'makesOrExpeditions.'], function () { ... });
  + Tương tự như Group Route trên, định nghĩa một nhóm route với tiền tố ‘makesOrExpeditions' và đặt tên nhóm là 'makesOrExpeditions'.
* Route::group(['prefix' => ‘wares’, 'as' => 'wares.'], function () { ... });
  + Tương tự như Group Route trên, định nghĩa một nhóm route với tiền tố ‘wares' và đặt tên nhóm là 'wares’.

## **Bước 6:** Xây dựng giao diện và cập nhật controller để xử lý CRUD

### **Tạo một layout chính.**

#### **File master**

File: **resources/views/layouts/master.blade.php**

| <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8">  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">  <title>@yield('title', 'Default Title')</title>  <meta *name*="csrf-token" *content*="{{ csrf\_token() }}">  <link *rel*="stylesheet" *href*="{{ asset('css/bootstrap.min.css') }}">  <link *rel*="stylesheet" *href*="{{ asset('fontawesome-free-6.4.0-web/css/all.min.css') }}">  <link *rel*="stylesheet" *href*="{{ asset('css/style.css') }}">  <link *href*="https://swisnl.github.io/jQuery-contextMenu/dist/jquery.contextMenu.css" *rel*="stylesheet"  *type*="text/css" />  @stack('css')  </head>  <body>  <header>  @include('layouts.header')  </header>  <main>  @yield('content')  </main>  <footer>  @include('layouts.footer')  </footer>  <script *src*="{{ asset('js/bootstrap.bundle.min.js') }}"></script>  <script *src*="{{ asset('js/jquery-3.7.0.min.js') }}"></script>  <script *src*="{{ asset('js/app.js') }}"></script>  <script *src*="https://swisnl.github.io/jQuery-contextMenu/dist/jquery.contextMenu.js" *type*="text/javascript"></script>  <script *src*="https://swisnl.github.io/jQuery-contextMenu/dist/jquery.ui.position.min.js"  *type*="text/javascript"></script>  @stack('javascript')  </body>  </html> |
| --- |

Trong file code trên:

* Thẻ <title> chứa tiêu đề của trang web, được lấy từ @yield('title', 'Default Title'). Nếu không có nội dung được cung cấp trong section @section('title'), tiêu đề mặc định sẽ là "Default Title".
* Thẻ <meta> chứa CSRF token, một giá trị bảo mật được sử dụng trong Laravel.
* Các liên kết <link> tới các tệp CSS như Bootstrap và Font Awesome để tạo kiểu cho trang web.
* @stack('css') chứa các đoạn mã CSS được đặt ở các vị trí khác nhau trong ứng dụng.
* Phần header, được chèn từ tệp header.blade.php trong thư mục layouts bằng cú pháp @include('layouts.header').
* Phần nội dung chính, được định nghĩa trong các trang con bằng cú pháp @yield('content').
* Phần footer, được chèn từ tệp footer.blade.php trong thư mục layouts bằng cú pháp @include('layouts.footer').
* Các tệp JavaScript được liên kết thông qua các thẻ <script> để cung cấp các chức năng tương tác và hiệu ứng cho trang web.
* @stack('javascript') chứa các đoạn mã JavaScript được đặt ở các vị trí khác nhau trong ứng dụng.

#### **File header**

File: **resources/views/layouts/header.blade.php**

| <nav *class*="navbar mb-3">  <div *class*="container-fluid">  <a *class*="navbar-brand pt-0" *href*="{{ route('index') }}">  <img *src*="{{ asset('images/logo.jpg') }}" *alt*="Logo" *class*="object-fit-cover" *height*="50">  </a>  @if (session('name'))  <div *class*="d-flex align-items-center justify-content-center gap-5">  <p *class*="m-0 text-primary-color">Xin chào {{ session('name') }}</p>  <a *href*="{{ route('logout') }}" *class*="btn text-primary-color">  Đăng xuất  </a>  </div>  @else  <a *href*="{{ route('login') }}" *class*="btn text-primary-color">Đăng nhập</a>  @endif  </div>  </nav> |
| --- |

Trong file code trên:

* Đoạn mã trên là một phần của một thanh điều hướng (navbar) trong một ứng dụng web. Nó có chức năng kiểm tra xem người dùng đã đăng nhập hay chưa và hiển thị các phần tương ứng:
  + Nếu người dùng chưa đăng nhập, thanh điều hướng sẽ hiển thị một nút "Đăng nhập". Khi người dùng nhấp vào nút này, một hộp thoại đăng nhập sẽ xuất hiện, cho phép người dùng nhập địa chỉ email và mật khẩu để đăng nhập vào hệ thống.
  + Nếu người dùng đã đăng nhập, thanh điều hướng sẽ hiển thị thông báo "Xin chào [Tên người dùng]" và một nút "Đăng xuất". Khi người dùng nhấp vào nút "Đăng xuất", họ sẽ được đăng xuất khỏi hệ thống.
* Ngoài ra, đoạn mã cũng cung cấp các tính năng bổ sung như:
  + Ngôn ngữ: Có hai nút "Tiếng Anh" và "Tiếng Việt" cho phép người dùng chọn ngôn ngữ hiển thị trang web.
  + Phần quên mật khẩu: Có một liên kết "Quên mật khẩu?" cho phép người dùng truy cập vào trang để khôi phục mật khẩu nếu cần.

#### **File footer**

<footer *class*="footer position-static bottom-0 start-0 end-0 bg-body-tertiary py-3 border-top border-primary-color">

<div *class*="container-fluid g-0 text-center">

<span *class*="text-body-secondary">

&copy; Copyright 2023 Ngan Giang company.

</span>

</div>

</footer>

#### **File CSS**

body{

display: flex;

flex-direction: column;

min-height: 100vh;

}

body main{

flex: 1;

}

textarea {

resize: none;

}

*.btn-primary-color*{

background-color: #2b4c72 !important;

color: #fff !important;

transition: 0.3s all linear;

}

*.bg-primary-color*{

background-color: #2b4c72 !important;

color: #fff !important;

}

*.btn-primary-color:hover*{

opacity: 0.9;

}

*.text-primary-color*{

color: #2b4c72 !important;

}

*.border-primary-color*{

border-color: #2b4c72 !important;

}

*.tabs* {

position: relative;

border-bottom: 2px solid rgba(43, 76, 114, 0.1);

}

*.tabs-line* {

position: absolute;

left: 0;

bottom: 0;

width: 0;

height: 2px;

background-color: #224874;

transform: all 0.2 ease;

}

*.tab-item* {

flex: 1;

font-size: 20px;

text-align: center;

color: #2b4c72;

background-color: #fff;

border-top-left-radius: 5px;

border-top-right-radius: 5px;

border-bottom: 5px solid transparent;

opacity: 0.6;

cursor: pointer;

transform: all 0.5s ease;

}

*.tab-item:hover* {

opacity: 1;

background-color: rgba(43, 76, 114, 0.05);

border-color: rgba(43, 76, 114, 0.1);

}

*.tab-item.active* {

opacity: 1;

}

*.tab-icon* {

font-size: 24px;

width: 32px;

position: relative;

top: 2px;

}

*.tab-pane* {

color: #333;

display: none;

}

*.tab-pane.active* {

display: block;

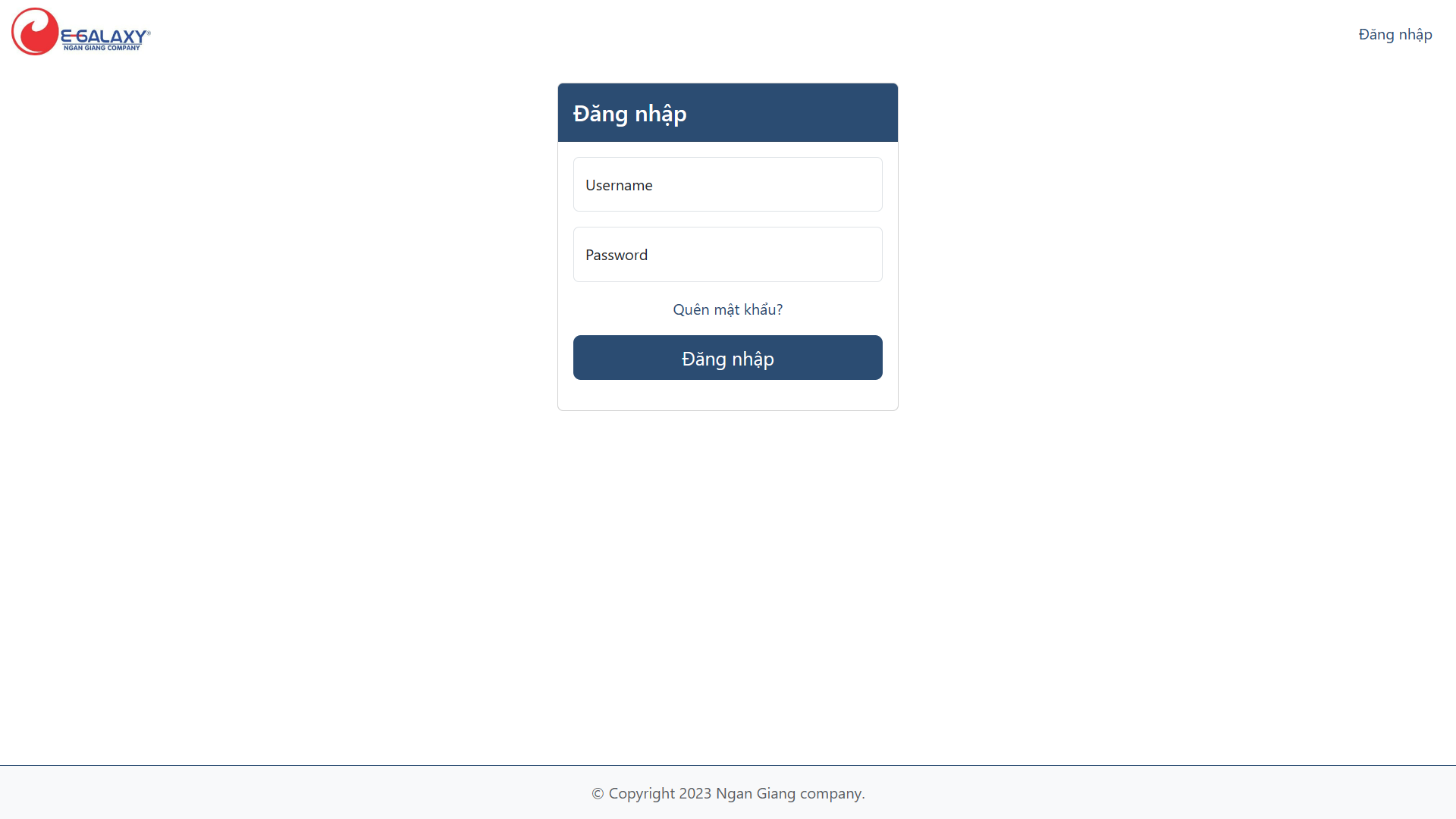
}

### **Giao diện chính**

File: **resources/views/index.blade.php**

* **Giao diện file**

****

****

* **Source code**

| @extends('layouts.master')  @section('title', 'Trang chủ')  @section('content')  <div *class*="container-fluid g-0">  <div *class*="row g-0">  <div *class*="col-md-12 g-0">  <div *class*="tabs d-flex justify-content-between">  <div *class*="tab-item active">  <h6>Quản lý</h6>  </div>  <div *class*="tab-item">  <h6>Dây chuyền xử lý sản xuất</h6>  </div>  <div *class*="tab-item">  <h6>Theo dõi</h6>  </div>  <div *class*="tab-item">  <h6>Quản trị hệ thống</h6>  </div>  <div *class*="tabs-line"></div>  </div>  </div>  </div>  </div>  <div *class*="container mt-3">  <div *class*="row">  <div *class*="col-md-12">  <div *class*="tab-content">  <div *class*="tab-pane active">  <div *class*="row">  <div *class*="col-md-6">  <div *class*="d-flex flex-column">  <a *href*="{{ route('makesOrExpeditions.index') }}" *class*="btn btn-primary-color my-1 w-50">  Quản lý đơn sản xuất và giao hàng  </a>  <a *href*="{{ route('orders.index') }}" *class*="btn btn-primary-color my-1 w-50">  Quản lý thùng hàng  </a>  <a *href*="{{ route('packs.index') }}" *class*="btn btn-primary-color my-1 w-50">Quản lý gói hàng</a>  <a *href*="{{ route('customers.index') }}" *class*="btn btn-primary-color my-1 w-50">  Quản lý khách hàng  </a>  <a *href*="{{ route('rawMaterials.index') }}" *class*="btn btn-primary-color my-1 w-50">  Quản lý nguyên liệu  </a>  <a *href*="{{ route('stations.index') }}" *class*="btn btn-primary-color my-1 w-50">Quản lý trạm</a>  <a *href*="{{ route('wares.index') }}" *class*="btn btn-primary-color my-1 w-50">Quản lý kho hàng</a>  </div>  </div>  <div *class*="col-md-6 text-center">  <h2 *class*="h2 text-primary-color">Quản lý sản xuất và giao hàng</h2>  </div>  </div>  </div>  <div *class*="tab-pane">  <div *class*="row">  <div *class*="d-flex flex-column">  <a *href*="{{ route('productionStationLines.create') }}" *class*="btn btn-primary-color w-25">  Thêm dây chuyền xử lý sản xuất mới  </a>  </div>  <div *class*="mt-2">  <table *class*="table table-striped w-100">  <thead>  <tr>  <th *class*="text-center" *scope*="col">#</th>  <th *scope*="col">Tên dây chuyền</th>  <th *scope*="col">Mô tả</th>  <th *scope*="col" *class*="text-center">Loại</th>  <th *scope*="col" *class*="text-center">Hoạt động</th>  </tr>  </thead>  <tbody *id*="table-data" *class*="table-group-divider ">  @foreach ($productionStationLines as $each)  <tr>  <th *class*="text-center" *scope*="col">{{ $each->Id\_ProdStationLine }}</th>  <td>{{ $each->Name\_ProdStationLine }}</td>  <td>{{ $each->Description }}</td>  <td *class*="text-center" *scope*="col">{{ $each->orderType->Name\_OrderType }}</td>  <td *class*="text-center">  <a *href*="{{ route('productionStationLines.show', $each) }}"  *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-eye"></i>  </a>  <a *href*="{{ route('productionStationLines.edit', $each) }}"  *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-pen-to-square"></i>  </a>  <a *href*="" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary"  *data-bs-toggle*="modal" *data-bs-target*="#i{{ $each->Id\_ProdStationLine }}">  <i *class*="fa-solid fa-trash"></i>  </a>  <div *class*="modal fade" *id*="i{{ $each->Id\_ProdStationLine }}" *tabindex*="-1"  *aria-labelledby*="exampleModalLabel" *aria-hidden*="true">  <div *class*="modal-dialog">  <div *class*="modal-content">  <div *class*="modal-header">  <h1 *class*="modal-title fs-5" *id*="exampleModalLabel">Xác nhận</h1>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal"  *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  Bạn có chắc chắn về việc xóa dây chuyền xử lý sản xuất này?  </div>  <div *class*="modal-footer">  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Hủy</button>  <form *action*="{{ route('productionStationLines.destroy', $each) }}" *method*="POST">  @csrf  @method('DELETE')  <button *type*="submit" *class*="btn btn-danger">Xóa</button>  </form>  </div>  </div>  </div>  </div>  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  </div>  </div>  <div *class*="tab-pane">  </div>  <div *class*="tab-pane">  </div>  </div>  </div>  </div>  </div>  @if (session('message') && session('type'))  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body bg-{{ session('type') }} d-flex align-items-center justify-content-between">  <div *class*=" d-flex justify-content-center align-items-center gap-2">  @if (session('type') == 'success')  <i *class*="fas fa-check-circle text-light fs-5"></i>  @elseif(session('type') == 'danger' || session('type') == 'warning')  <i *class*="fas fa-xmark-circle text-light fs-5"></i>  @elseif(session('type') == 'info' || session('type') == 'secondary')  <i *class*="fas fa-info-circle text-light fs-5"></i>  @endif  <h6 *class*="h6 text-white m-0">{{ session('message') }}</h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endif  @endsection  @push('javascript')  <script *type*="text/javascript">  $(document).ready(function () {  const tabs = $(".tab-item");  const panes = $(".tab-pane");  const tabActive = $(".tab-item.active");  const line = $(".tabs-line");  line.css({  left: tabActive.position().left + "px",  width: tabActive.outerWidth() + "px"  });  tabs.each(function (*index*) {  let tab = $(*this*);  let pane = panes.eq(*index*);  tab.on("click", function () {  $(".tab-item.active").removeClass("active");  $(".tab-pane.active").removeClass("active");  line.css({  left: tab.position().left + "px",  width: tab.outerWidth() + "px"  });  tab.addClass("active");  pane.addClass("active");  });  });  const toastLiveExample = $('#liveToast');  const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));  toastBootstrap.show();  })  let selectElement = $("#Id\_Station");  $(selectElement).on("change", function () {  let id = $(*this*).val();  $.ajax({  method: "POST",  data: {  id: id,  \_token: $('meta[name="csrf-token"]').attr("content"),  },  url: "/",  *// url: url,*  success: function (*response*) {  let data = *response*.data;  console.log(data);  $("#table-data").empty();  $.each(data, function (*key*, *value*) {  let id = *value*.Id\_ProdStationLine;  let name = *value*.Name\_ProdStationLine;  let description = *value*.Description;  let showUrl =  "{{ route('productionStationLines.show', ':id') }}".replace(  ":id",  id  );  let editUrl =  "{{ route('productionStationLines.edit', ':id') }}".replace(  ":id",  id  );  let deleteModalId = "deleteModal-" + id;  let deleteUrl =  "{{ route('productionStationLines.destroy', ':id') }}".replace(  ":id",  id  );  let htmls = `<tr>  <th class="text-center" scope="row">${id}</th>  <td>${name}</td>  <td>${description}</td>  <td>  <a href="${showUrl}" class="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i class="fa-solid fa-eye"></i>  </a>  <a href="${editUrl}" class="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i class="fa-solid fa-pen-to-square"></i>  </a>  <button type="button" class="btn btn-sm btn-outline-light text-primary-color border-secondary" data-bs-toggle="modal" data-bs-target="#${deleteModalId}">  <i class="fa-solid fa-trash"></i>  </button>  <div class="modal fade" id="${deleteModalId}" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog">  <div class="modal-content">  <div class="modal-header">  <h1 class="modal-title fs-5" id="exampleModalLabel">Xác nhận</h1>  <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>  </div>  <div class="modal-body">  Bạn có chắc chắn về việc xóa dây chuyền xử lý sản xuất này?  </div>  <div class="modal-footer">  <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Hủy</button>  <form action="${deleteUrl}" method="POST">  @csrf  @method('DELETE')  <button type="submit" class="btn btn-danger">Xóa</button>  </form>  </div>  </div>  </div>  </div>  </td>  </tr>`;  $("#table-data").append(htmls);  });  },  });  });  </script>  @endpush |
| --- |

Trong file code trên:

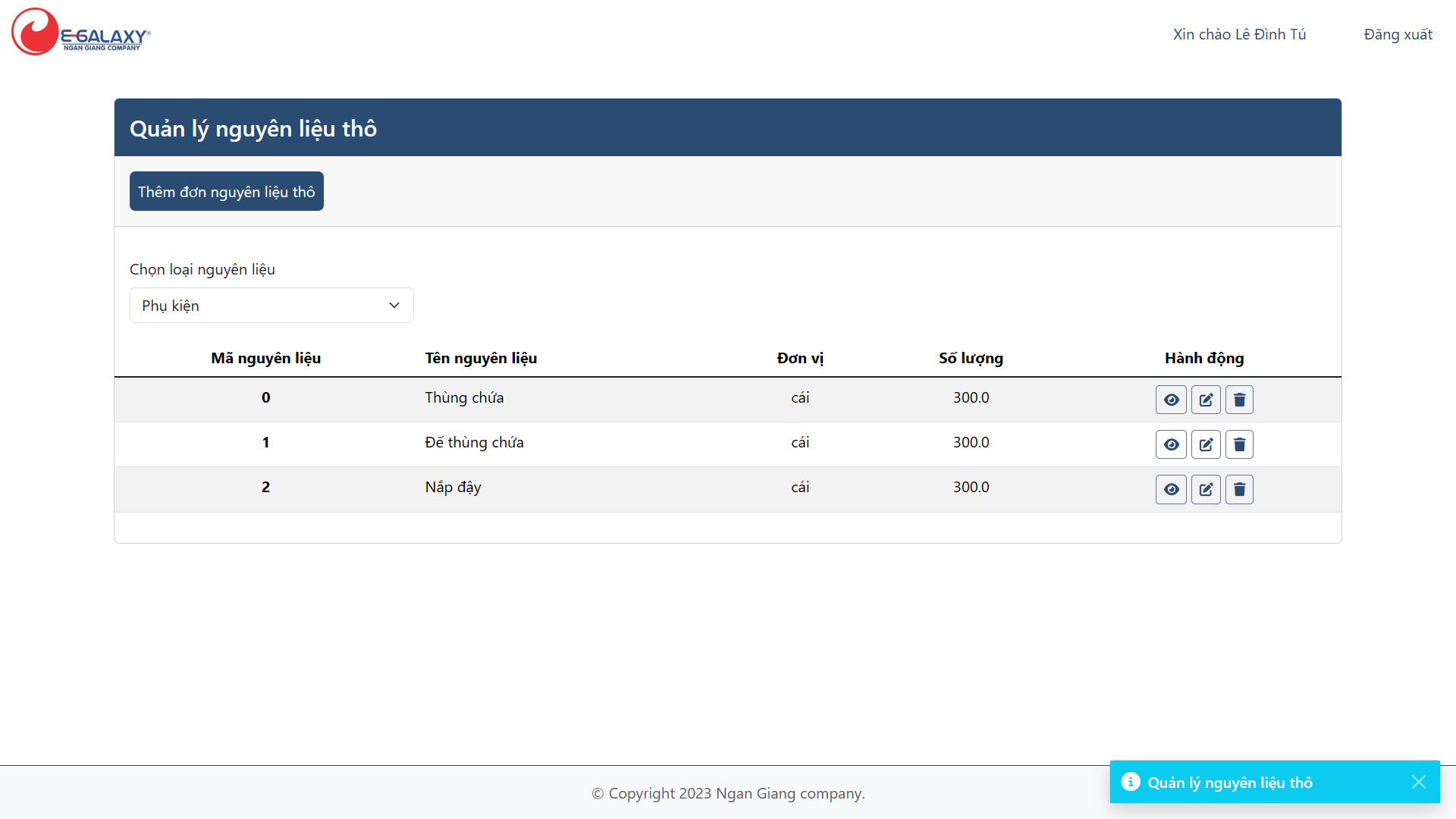
* @extends('layouts.app') chỉ định rằng view này sẽ kế thừa từ layout app.blade.php.
* @section('title', 'Welcome Page') đặt tiêu đề cho trang. Nếu không có tiêu đề được đặt, thì tiêu đề mặc định (trong trường hợp này là "Default Title") sẽ được sử dụng.
* @section('content') ... @endsection định nghĩa nội dung cho vùng @yield('content') trong layout chính.
* push(‘javascript’): Điều hướng đến app.js
* Giao diện này sẽ kế thừa giao diện header, footer được tạo riêng ở layouts/header.blade.php và footer.blade.php
* File JavaScript: app.js **(public\js\app.js)**

| $(document).ready(function () {  var station\_group = $(".station-group");  var station\_select = $(".station-select");  var station\_label = $(".station-label");  var station\_start = $("#Station\_Start");  var station\_end = $("#Station\_End");  *// Khi thay đổi trạm đầu*  station\_start.on("change", function () {  hideAllStation(station\_group);  if ($(*this*).val() == 401) {  *// trạm bắt đầu bằng 401*  *// Hiển thị lại lựa chọn 406, 407, 409 của select trạm cuối*  showStation($("#Station\_End > option[value='406']"));  showStation($("#Station\_End > option[value='407']"));  showStation($("#Station\_End > option[value='409']"));  *// Đặt lại trạm cuối = 406*  station\_end.val(406);  station\_end.change();  } else if ($(*this*).val() == 406) {  *// trạm đầu bằng 406*  *// Ẩn lựa chọn 406 của select trạm cuối*  hideStation($("#Station\_End > option[value='406']"));  *// Đặt lại trạm cuối bằng 407*  station\_end.val(407);  station\_end.change();  *// Hiển thị lại lựa chọn 407, 409 cho trạm cuối*  showStation($("#Station\_End > option[value='407']"));  showStation($("#Station\_End > option[value='409']"));  } else {  *// trạm đầu bằng 409*  *// Ẩn lựa chọn 406, 407, 409*  hideStation($("#Station\_End > option[value='406']"));  hideStation($("#Station\_End > option[value='407']"));  hideStation($("#Station\_End > option[value='409']"));  *// Đặt lại trạm cuối bằng 412*  station\_end.val(412);  station\_end.change();  }  });  *// Khi thay đổi trạm cuối*  station\_end.on("change", function () {  *// Ẩn tất cả các station*  let count = 2;  hideAllStation(station\_group);  if (station\_start.val() == 401) {  if ($(*this*).val() == 406) {  *// trạm đầu là 401, trạm cuối là 406*  for (let i = 0; i < 3; i++) {  showStation(station\_group.eq(i));  }  disabledStation(station\_select.eq(2));  } else if ($(*this*).val() == 407) {  *// trạm dầu là 401, trạm cuối là 407*  for (let i = 0; i < 3; i++) {  showStation(station\_group.eq(i));  }  station\_select.eq(2).change();  } else if ($(*this*).val() == 409) {  *// trạm đầu là 401, trạm cuối là 409*  for (let i = 0; i < 6; i++) {  showStation(station\_group.eq(i));  }  station\_select.eq(2).change();  } else {  *// trạm đầu là 401, trạm cuối là 412*  for (let i = 0; i < 9; i++) {  showStation(station\_group.eq(i));  }  station\_select.eq(2).change();  station\_select.eq(3).change();  }  } else if (station\_start.val() == 406) {  hideStation($("#Station\_End > option[value='406']"));  if (station\_end.val() == 407) {  hideAllStation(station\_group);  } else {  if (station\_end.val() == 409) {  *// trạm cuối = 409*  for (let i = 3; i < 6; i++) {  showStation(station\_group.eq(i));  station\_label.eq(i).html("Chọn trạm thứ " + count++);  }  unDisabledStation(station\_select.eq(2));  disabledStation(station\_select.eq(4));  disabledStation(station\_select.eq(5));  } else {  *// trạm cuối = 412*  for (let i = 3; i < 9; i++) {  showStation(station\_group.eq(i));  station\_label.eq(i).html("Chọn trạm thứ " + count++);  }  station\_select.eq(3).change();  }  }  } else if (station\_start.val() == 409) {  if ($(*this*).val() == 412) {  *// Trạm dầu là 409, trạm cuối là 412*  for (let i = 6; i < 9; i++) {  showStation(station\_group.eq(i)); *// Hiển thị station*  station\_label.eq(i).html("Chọn trạm thứ " + count++); *// Đổi tên station*  }  disabledStation(station\_select.eq(6)); *// không cho lựa chọn eq6*  }  }  });  station\_select.eq(2).on("change", function () {  *// Sự kiện thay đổi trạm thứ 4*  *// trạm 4 có 4 option là 406 và 407*  if (station\_end.val() == 407 && $(*this*).val() == station\_end.val()) {  *// Trạm 4 = 407 = trạm cuối*  *// console.log('this is 407 end');*  hideStation(station\_group.eq(3));  } else {  *// Trạm 4 = 406 != trạm cuối*  if ($(*this*).val() == 406) {  *// console.log('this is 406 not end');*  showStation(station\_group.eq(3));  if (station\_end.val() == 407) {  disabledStation(station\_select.eq(3));  *// console.log('1');*  unDisabledStation(station\_select.eq(2));  } else if (station\_end.val() == 409) {  station\_select.eq(3).val(407);  station\_select.eq(4).val(408);  station\_select.eq(5).val(409);  unDisabledStation(station\_select.eq(2));  unDisabledStation(station\_select.eq(3));  showStation(station\_group.eq(5));  disabledStation(station\_select.eq(4));  disabledStation(station\_select.eq(5));  *// console.log('2');*  } else {  *// console.log('3');*  unDisabledStation(station\_select.eq(2));  unDisabledStation(station\_select.eq(3));  }  } else {  *// console.log('this is 407 not end')*  *// Trạm 4 = 407 != trạm cuối*  if (station\_end.val() == 409) {  *// trạm 4 = 407, trạm cuối = 409*  station\_select.eq(3).val(408);  station\_select.eq(4).val(409);  disabledStation(station\_select.eq(3));  disabledStation(station\_select.eq(4));  hideStation(station\_group.eq(5));  *// console.log('1');*  } else {  *// trạm 4 = 407, trạm cuối = 412*  *// console.log('2');*  station\_select.eq(3).val(408);  disabledStation(station\_select.eq(3));  *// Sửa selectbox trạm 6*  hideStation(  $(".station-select:eq(4) > option[value='408']")  );  station\_select.eq(4).val(409);  station\_select.eq(5).val(410);  station\_select.eq(6).val(411);  station\_select.eq(7).val(412);  unDisabledStation(station\_select.eq(4));  disabledStation(station\_select.eq(5));  disabledStation(station\_select.eq(6));  disabledStation(station\_select.eq(7));  hideStation(station\_group.eq(8));  }  }  }  });  station\_select.eq(3).on("change", function () {  if ($(*this*).val() == 407) {  *// Trạm 5 = 407*  *// console.log('Trạm 5 407');*  if (station\_end.val() == 407) {  *// Trạm 5 = trạm cuối*  *// console.log('1');*  disabledStation($(*this*));  } else if (station\_end.val() == 412) {  *// Trạm 5 khác trạm cuối (408)*  *// console.log('2');*  station\_select.eq(4).val(408);  showStation(station\_group.eq(5));  unDisabledStation(station\_select.eq(5));  disabledStation(station\_select.eq(4));  disabledStation(station\_select.eq(6));  hideStation($(".station-select:eq(5) > option[value='411']"));  } else {  *// trạm cuối = 409*  *// console.log(3);*  station\_select.eq(4).val(408);  showStation(station\_group.eq(5));  }  } else {  *// Trạm 5 = 408*  *// console.log('Trạm 5 408');*  if (station\_end.val() == 409) {  *// Trạm cuối = 409*  station\_select.eq(4).val(409);  hideStation(station\_group.eq(5));  *// console.log('1');*  } else {  *// Trạm cuối = 412*  *// console.log('2');*  station\_select.eq(4).val(409);  station\_select.eq(5).val(410);  station\_select.eq(6).val(411);  station\_select.eq(7).val(412);  disabledStation(station\_select.eq(5));  unDisabledStation(station\_select.eq(4));  hideStation($(".station-select:eq(4) > option[value='408']"));  hideStation(station\_group.eq(8));  }  }  });  station\_select.eq(4).on("change", function () {  *// trạm 6*  if ($(*this*).val() == 409) {  *// Trạm 6 = 409*  if (station\_end.val() == 409) {  *// Trạm 6 = 409 và trạm cuối == 409*  *// console.log('Trạm 6 end 409')*  } else {  *// Trạm 6 = 409 và trạm cuối != 409*  station\_select.eq(5).val(410);  station\_select.eq(6).val(411);  showStation(station\_group.eq(7));  }  } else {  *// Trạm 6 = 410*  station\_select.eq(5).val(411);  station\_select.eq(6).val(412);  hideStation(station\_group.eq(7));  }  });  station\_select.eq(5).on("change", function () {  *// trạm 7*  if ($(*this*).val() == 409) {  if (station\_end.val() == 409) {  *// Trạm 7 = 409, là trạm cuối*  *// console.log('trạm 7 = 409, trạm cuối');*  } else {  *// Trạm 7 = 409, không phải trạm cuối*  *// console.log('trạm 7 = 409, không phải trạm cuối');*  station\_select.eq(6).val(410);  station\_select.eq(7).val(411);  showStation(station\_group.eq(8));  }  } else {  *// trạm 7 = 410*  station\_select.eq(6).val(411);  station\_select.eq(7).val(412);  hideStation(station\_group.eq(8));  }  });  *// Reset nút station\_start, station\_end khi load trang*  station\_start.trigger("change");  station\_end.trigger("change");  });  function disabledStation(*station\_disable*) {  station\_disable.attr("disabled", "disabled");  }  function unDisabledStation(*station\_disable*) {  station\_disable.removeAttr("disabled");  }  function hideStation(*station\_show*) {  station\_show.hide();  }  function showStation(*station\_hide*) {  station\_hide.show();  }  function hideAllStation(*station\_group*) {  station\_group.each(function () {  $(*this*).hide();  });  } |
| --- |

### **Giao diện quản lý nguyên vật liệu**

#### **Giao diện hiển thị nguyên vật liệu**

File: **resources\views\rawMaterials\index.blade.php**

****

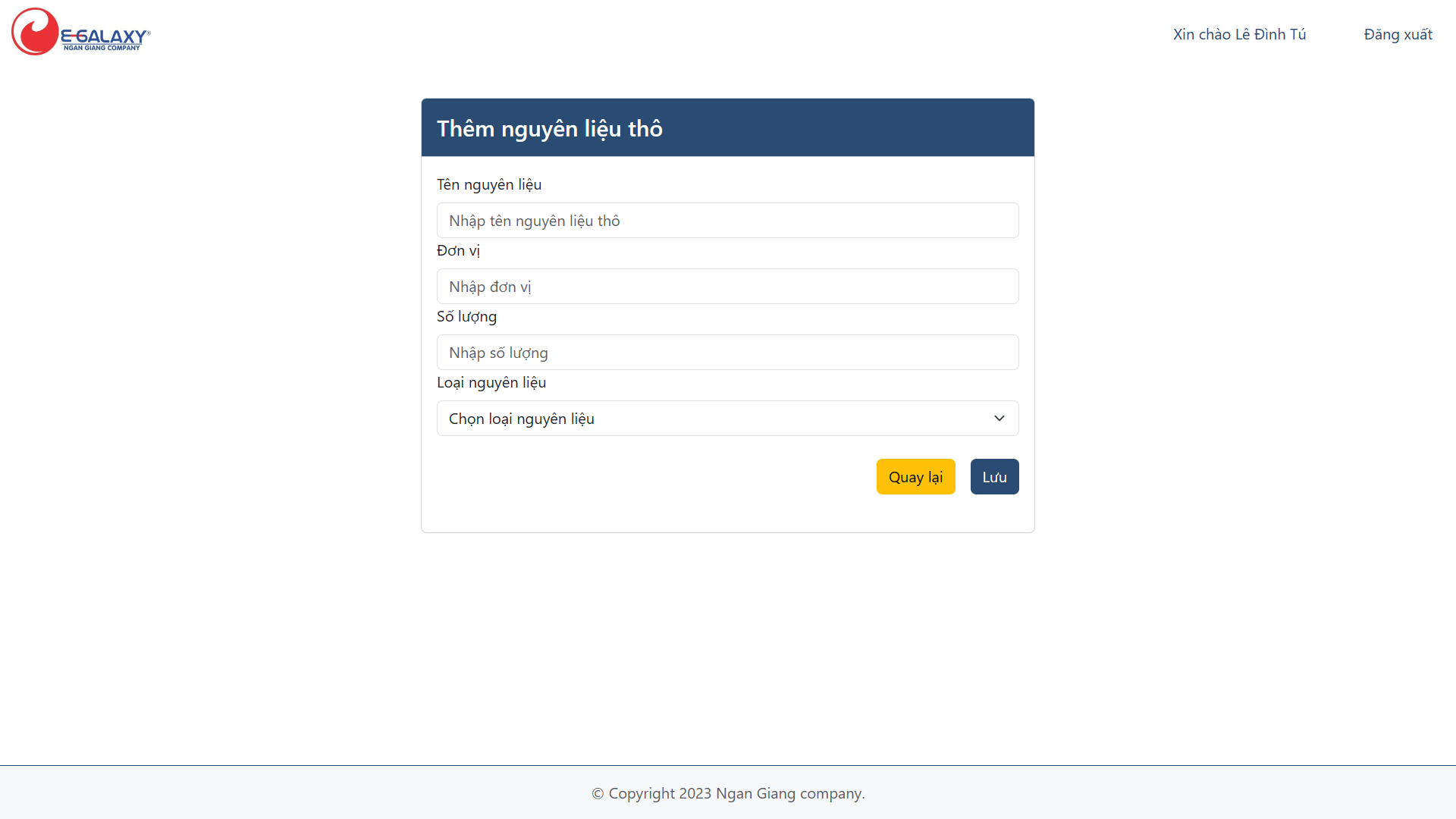
| @extends('layouts.master')  @section('title', 'Quản lý nguyên liệu thô')  @section('content')  <div *class*="container">  <div *class*="row">  <div *class*="col-md-12">  <div *class*="card mt-3">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Quản lý nguyên liệu thô</h4>  <a *href*="{{ route('rawMaterials.create')}}" *class*="btn btn-primary-color text-white p-2 my-3 ms-3">  Thêm đơn nguyên liệu thô  </a>  </div>  <div *class*="card-body px-0">  <div *class*="ms-3 my-3" *style*="width: 300px;">  <label *for*="FK\_Id\_RawMaterialType" *class*="form-label">Chọn loại nguyên liệu</label>  <select *name*="FK\_Id\_RawMaterialType" *class*="form-select">  @foreach($types as $each)  <option *value*="{{ $each->id }}">{{ $each->name }}</option>  @endforeach  </select>  </div>  <table *class*="table table-striped w-100">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Mã nguyên liệu</th>  <th *scope*="col">Tên nguyên liệu</th>  <th *class*="text-center" *scope*="col">Đơn vị</th>  <th *class*="text-center" *scope*="col">Số lượng</th>  <th *class*="text-center" *scope*="col">Hành động</th>  </tr>  </thead>  <tbody *id*="table-data" *class*="table-group-divider">  </tbody>  </table>  </div>  </div>  </div>  </div>  </div>  @if(session('message') && session('type'))  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body bg-{{ session('type') }} d-flex align-items-center justify-content-between">  <div *class*=" d-flex justify-content-center align-items-center gap-2">  @if(session('type') == 'success')  <i *class*="fas fa-check-circle text-light fs-5"></i>  @elseif(session('type') == 'danger' || session('type') == 'warning')  <i *class*="fas fa-xmark-circle text-light fs-5"></i>  @elseif(session('type') == 'info' || session('type') == 'secondary')  <i *class*="fas fa-info-circle text-light fs-5"></i>  @endif  <h6 *class*="h6 text-white m-0">{{ session('message') }}</h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endif  @endsection  @push('javascript')  <script>  const toastLiveExample = $('#liveToast');  const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));  toastBootstrap.show();  let selectElement = $("select[name='FK\_Id\_RawMaterialType']");  $(selectElement).on('change', function () {  let id = $(*this*).val();  let token = $('meta[name="csrf-token"]').attr('content');  $.ajax({  url: "{{ route('rawMaterials.showMaterials') }}",  method: 'POST',  dataType: 'json',  data: {  id: id,  \_token: token  },  success: function (*response*) {  let data = *response*.data;  $("#table-data").empty(); *// Làm rỗng phần tử HTML có id "table-data"*  $.each(data, function (*key*, *value*) {  let id = *value*.Id\_RawMaterial;  let materialName = *value*.Name\_RawMaterial;  let unit = *value*.Unit;  let count = *value*.count;  let showUrl = "{{ route('rawMaterials.show', ':id') }}".replace(':id', id);  let editUrl = "{{ route('rawMaterials.edit', ':id') }}".replace(':id', id);  let deleteModalId = "deleteModal-" + id;  let deleteUrl = "{{ route('rawMaterials.destroy', ':id') }}".replace(':id', id);  let htmls =  `<tr>  <th class="text-center" scope="row">${id}</th>  <td>${materialName}</td>  <td class="text-center">${unit}</td>  <td class="text-center">${count}</td>  <td class="text-center">  <a href="${showUrl}" class="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i class="fa-solid fa-eye"></i>  </a>  <a href="${editUrl}" class="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i class="fa-solid fa-pen-to-square"></i>  </a>  <button type="button" class="btn btn-sm btn-outline-light text-primary-color border-secondary" data-bs-toggle="modal" data-bs-target="#${deleteModalId}">  <i class="fa-solid fa-trash"></i>  </button>  <div class="modal fade" id="${deleteModalId}" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog">  <div class="modal-content">  <div class="modal-header">  <h1 class="modal-title fs-5" id="exampleModalLabel">Xác nhận</h1>  <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>  </div>  <div class="modal-body">  Bạn có chắc chắn về việc xóa nguyên liệu thô này?  </div>  <div class="modal-footer">  <button type="button" class="btn btn-warning" data-bs-dismiss="modal">Hủy</button>  <form action="${deleteUrl}" method="POST">  @csrf  @method('DELETE')  <button type="submit" class="btn btn-danger">Xóa</button>  </form>  </div>  </div>  </div>  </div>  </td>  </tr>`;  $("#table-data").append(htmls);  });  }  });  });  var firstOptionValue = $(selectElement).val();  *// Gán giá trị cho phần tử select*  $(selectElement).val(firstOptionValue);  *// Gọi sự kiện change để hiển thị dữ liệu*  $(selectElement).change();  </script>  @endpush |
| --- |

Trong file code trên:

* Sẽ có một thẻ select <option> để cho người dùng chọn ‘Loại nguyên vật liệu thô’, khi người dùng chọn thì sẽ có dữ liệu từ cơ sở dữ liệu (Sử dụng Ajax kết hợp với Jquery để lấy dữ liệu) trả về tương ứng với loại hàng đó.

#### **Giao diện thêm nguyên vật liệu**

File: **resources\views\rawMaterials\create.blade.php**

****

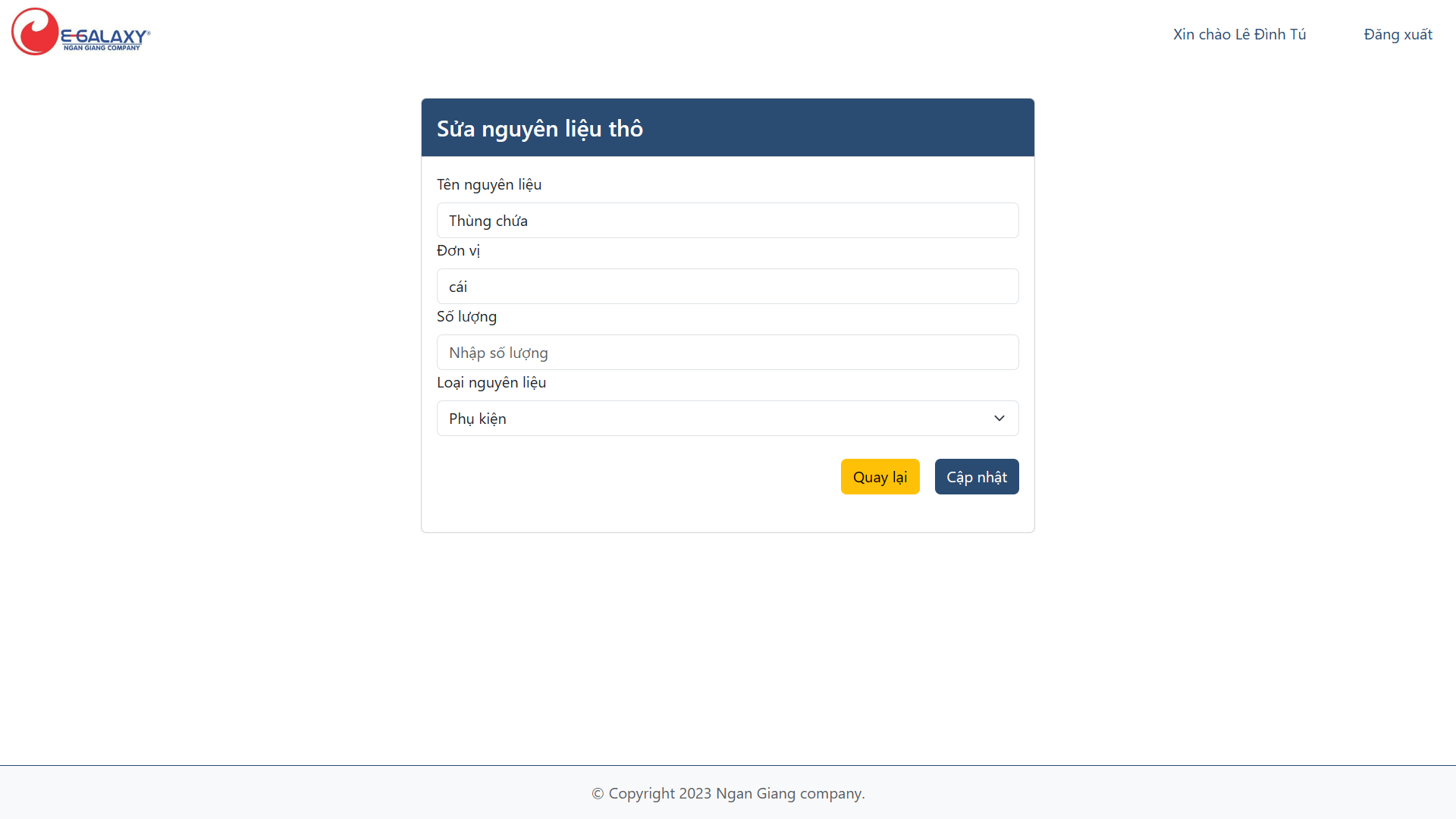
| @extends('layouts.master')  @section('title', 'Tạo nguyên liệu thô')  @section('content')  <div *class*="container">  <div *class*="row">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-50 mt-3">  <div *class*="card">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thêm nguyên liệu thô</h4>  </div>  <div *class*="card-body">  <form *method*="POST" *action*="{{ route('rawMaterials.store') }}">  @csrf  <div *class*="form-group">  <label *for*="Name\_RawMaterial" *class*="form-label">Tên nguyên liệu</label>  <input *type*="text" *name*="Name\_RawMaterial" *id*="Name\_RawMaterial" *placeholder*="Nhập tên nguyên liệu thô"  *class*="form-control{{ $errors->has('Name\_RawMaterial') ? ' is-invalid' : '' }}">  @if ($errors->has('Name\_RawMaterial'))  <span *class*="text-danger">  {{ $errors->first('Name\_RawMaterial') }}  </span>  @endif  </div>  <div *class*="form-group">  <label *for*="Unit" *class*="form-label">Đơn vị</label>  <input *type*="text" *name*="Unit" *id*="Unit" *placeholder*="Nhập đơn vị"  *class*="form-control{{ $errors->has('Unit') ? ' is-invalid' : '' }}">  @if ($errors->has('Unit'))  <span *class*="text-danger">  {{ $errors->first('Unit') }}  </span>  @endif  </div>  <div *class*="form-group">  <label *for*="Count" *class*="form-label">Số lượng</label>  <input *type*="number" *name*="Count" *id*="Count" *placeholder*="Nhập số lượng"  *class*="form-control{{ $errors->has('Count') ? ' is-invalid' : '' }}">  @if ($errors->has('Count'))  <span *class*="text-danger">  {{ $errors->first('Count') }}  </span>  @endif  </div>  <div *class*="form-group">  <label *for*="FK\_Id\_RawMaterialType" *class*="form-label">Loại nguyên liệu</label>  <select *name*="FK\_Id\_RawMaterialType"  *class*="form-select{{ $errors->has('FK\_Id\_RawMaterialType') ? ' is-invalid' : '' }}">  <option *value*="default">Chọn loại nguyên liệu</option>  @foreach($data as $each)  <option *value*="{{ $each->id }}">{{ $each->Name\_RawMaterialType }}</option>  @endforeach  </select>  @if ($errors->has('FK\_Id\_RawMaterialType'))  <span *class*="text-danger">  {{ $errors->first('FK\_Id\_RawMaterialType') }}  </span>  @endif  </div>  <div *class*="d-flex justify-content-end my-4 gap-3">  <a *href*="{{ route('rawMaterials.index') }}" *class*="btn btn-warning">Quay lại</a>  <button *type*="submit" *class*="btn btn-primary-color">Lưu</button>  </div>  </form>  </div>  </div>  </div>  </div>  </div>  </div>  @endsection |
| --- |

Trong file code trên:

* Sẽ có một form cho người dùng nhập dữ liệu, nếu nhập sai thì sẽ bị validated, còn nếu nhập đúng thì dữ liệu sẽ được thêm vào máy chủ.

#### **Giao diện sửa nguyên vật liệu**

File: **resources\views\rawMaterials\edit.blade.php**

****

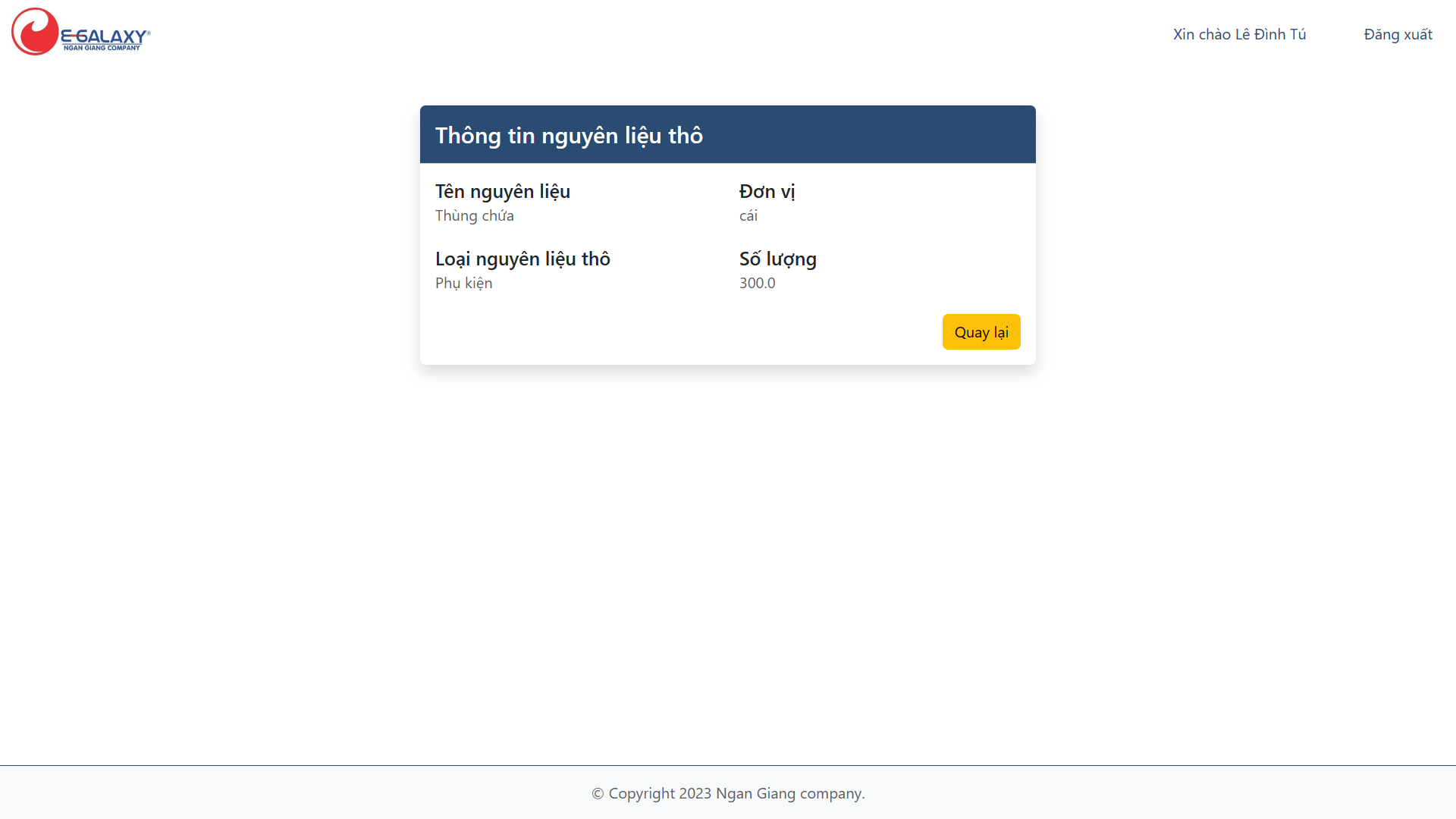
| @extends('layouts.master')  @section('title', 'Sửa nguyên liệu thô')  @section('content')  <div *class*="container">  <div *class*="row">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-50 mt-3">  <div *class*="card">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Sửa nguyên liệu thô</h4>  </div>  <div *class*="card-body">  <form *method*="POST" *action*="{{ route('rawMaterials.update', $material->Id\_RawMaterial ) }}">  @csrf  @method('PUT')  <div *class*="form-group">  <label *for*="Name\_RawMaterial" *class*="form-label">Tên nguyên liệu</label>  <input *type*="text" *name*="Name\_RawMaterial" *id*="Name\_RawMaterial" *placeholder*="Nhập tên nguyên liệu thô"  *value*="{{ $material->Name\_RawMaterial}}"  *class*="form-control{{ $errors->has('Name\_RawMaterial') ? ' is-invalid' : '' }}">  @if ($errors->has('Name\_RawMaterial'))  <span *class*="text-danger">  {{ $errors->first('Name\_RawMaterial') }}  </span>  @endif  </div>  <div *class*="form-group">  <label *for*="Unit" *class*="form-label">Đơn vị</label>  <input *type*="text" *name*="Unit" *id*="Unit" *placeholder*="Nhập đơn vị" *value*="{{ $material->Unit}}"  *class*="form-control{{ $errors->has('Unit') ? ' is-invalid' : '' }}">  @if ($errors->has('Unit'))  <span *class*="text-danger">  {{ $errors->first('Unit') }}  </span>  @endif  </div>  <div *class*="form-group">  <label *for*="count" *class*="form-label">Số lượng</label>  <input *type*="number" *name*="count" *id*="count" *placeholder*="Nhập số lượng" *value*="{{ $material->count}}"  *class*="form-control{{ $errors->has('count') ? ' is-invalid' : '' }}">  @if ($errors->has('count'))  <span *class*="text-danger">  {{ $errors->first('count') }}  </span>  @endif  </div>  <div *class*="form-group">  <label *for*="FK\_Id\_RawMaterialType" *class*="form-label">Loại nguyên liệu</label>  <select *name*="FK\_Id\_RawMaterialType"  *class*="form-select{{ $errors->has('FK\_Id\_RawMaterialType') ? ' is-invalid' : '' }}">  @foreach($data as $each)  @if($each->id == $material->FK\_Id\_RawMaterialType)  <option *value*="{{ $each->id }}" *selected*>{{ $each->Name\_RawMaterialType }}  </option>  @else  <option *value*="{{ $each->id }}">{{ $each->Name\_RawMaterialType }}</option>  @endif  @endforeach  </select>  @if ($errors->has('FK\_Id\_RawMaterialType'))  <span *class*="text-danger">  {{ $errors->first('FK\_Id\_RawMaterialType') }}  </span>  @endif  </div>  <div *class*="d-flex justify-content-end my-4 gap-3">  <a *href*="{{ route('rawMaterials.index') }}" *class*="btn btn-warning">Quay lại</a>  <button *type*="submit" *class*="btn btn-primary-color">Cập nhật</button>  </div>  </form>  </div>  </div>  </div>  </div>  </div>  </div>  @endsection |
| --- |

Trong file code trên:

* Sẽ có một form cho người dùng sửa dữ liệu, nếu nhập lỗi thì sẽ bị validated, còn nếu nhập đúng thì dữ liệu sẽ được sửa trong máy chủ.

#### **Giao diện hiển thị chi tiết nguyên vật liệu**

File: **resources\views\rawMaterials\show.blade.php**

****

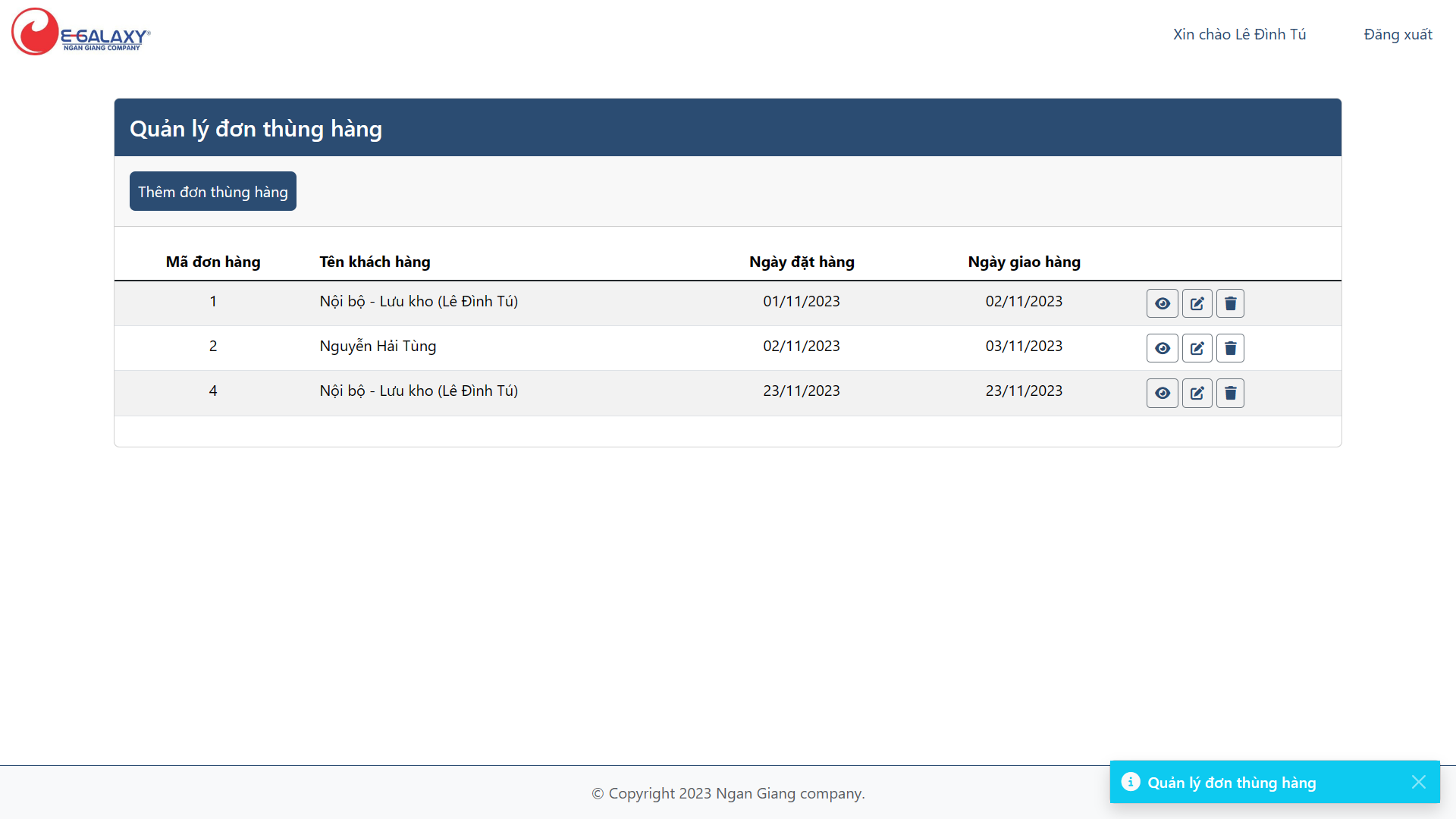
| @extends('layouts.master')  @section('title', 'Xem chi tiết nguyên liệu thô')  @section('content')  <div *class*="container my-4">  <div *class*="row">  <div *class*="col-md-12 d-flex align-items-center justify-content-center">  <div *class*="card border-0 shadow overflow-hidden" *style*="width: 650px">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin nguyên liệu thô</h4>  </div>  <div *class*="row g-0">  <div *class*="card-body d-flex flex-column justify-content-between h-100">  <div *class*="row">  <div *class*="col-md-6 mb-4">  <h5 *class*="h5 fw-medium mb-1">Tên nguyên liệu</h5>  <h6 *class*="h6 text-muted fw-normal m-0">{{ $material->Name\_RawMaterial }}</h6>  </div>  <div *class*="col-md-6 mb-4">  <h5 *class*="h5 fw-medium mb-1">Đơn vị</h5>  <h6 *class*="h6 text-muted fw-normal m-0">{{ $material->Unit }}</h6>  </div>  <div *class*="col-md-6 mb-4">  <h5 *class*="h5 fw-medium mb-1">Loại nguyên liệu thô</h5>  <h6 *class*="h6 text-muted fw-normal m-0">{{ $material->types->Name\_RawMaterialType }}</h6>  </div>  <div *class*="col-md-6 mb-4">  <h5 *class*="h5 fw-medium mb-1">Số lượng</h5>  <h6 *class*="h6 text-muted fw-normal m-0">{{ $material->Count }}</h6>  </div>  </div>  <div *class*="row">  <div *class*="col-md-12">  <div *class*="d-flex justify-content-end align-items-center gap-2">  <a *href*="{{ route('rawMaterials.index') }}" *class*="btn btn-warning">Quay lại</a>  </div>  </div>  </div>  </div>  </div>  </div>  </div>  </div>  </div>  @endsection |
| --- |

* Trong file code trên: Sẽ có một giao diện hiển thị đầy đủ thông tin của loại nguyên vật liệu.

### **Giao diện quản lý đơn thùng hàng**

#### **Giao diện hiển thị đơn thùng hàng**

File: **resources\views\simples\index.blade.php**

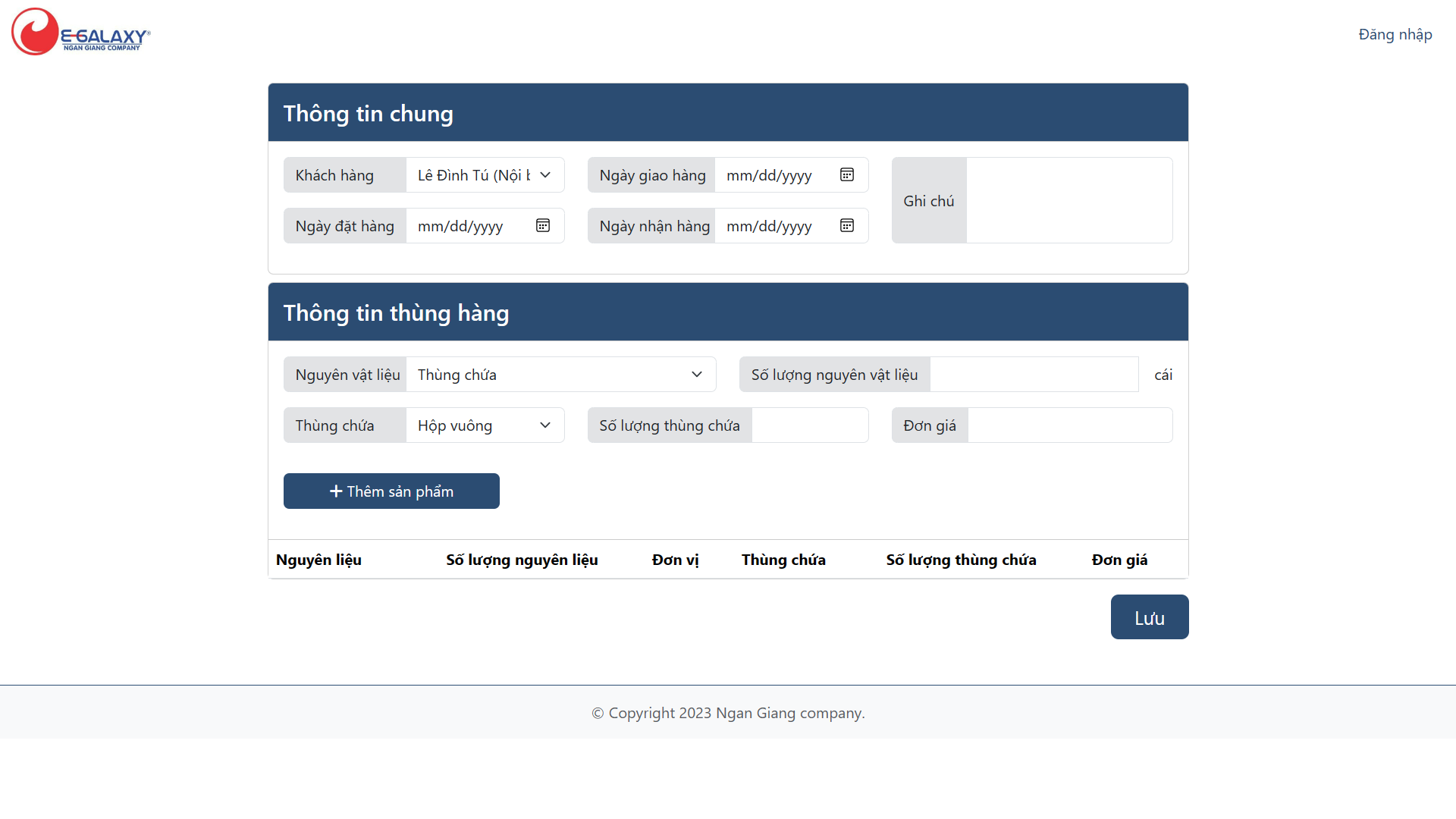


| @extends('layouts.master')  @section('title', 'Quản lý đơn thùng hàng')  @section('content')  <div *class*="container">  <div *class*="row">  <div *class*="col-md-12">  <div *class*="card mt-3">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Quản lý đơn thùng hàng</h4>  <a *href*="{{ route('orders.create')}}" *class*="btn btn-primary-color text-white p-2 my-3 ms-3">  Thêm đơn thùng hàng  </a>  </div>  <div *class*="card-body px-0">  <table *class*="table table-striped w-100">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Mã đơn hàng</th>  <th *scope*="col">Tên khách hàng</th>  <th *class*="text-center" *scope*="col">Ngày đặt hàng</th>  <th *class*="text-center" *scope*="col">Ngày giao hàng</th>  </tr>  </thead>  <tbody *id*="table-data" *class*="table-group-divider">  @foreach($data as $each)  <tr>  <td *class*="text-center">{{ $each->Id\_Order }}</td>  <td>{{ $each->customer->Name\_Customer}}</td>  <td *class*="text-center">{{ $each->order\_date}}</td>  <td *class*="text-center">{{ $each->delivery\_date}}</td>  <td>  <a *href*="{{ route('orders.show', $each) }}"  *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-eye"></i>  </a>  <a *href*="{{ route('orders.edit', $each) }}"  *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-pen-to-square"></i>  </a>  <button *type*="button" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary"  *data-bs-toggle*="modal" *data-bs-target*="#deleteOrder-{{$each->Id\_Order}}">  <i *class*="fa-solid fa-trash"></i>  </button>  <div *class*="modal fade" *id*="deleteOrder-{{$each->Id\_Order}}" *tabindex*="-1"  *aria-labelledby*="exampleModalLabel" *aria-hidden*="true">  <div *class*="modal-dialog">  <div *class*="modal-content">  <div *class*="modal-header">  <h1 *class*="modal-title fs-5" *id*="exampleModalLabel">Xác nhận</h1>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  Bạn có chắc chắn về việc sản phẩm này  </div>  <div *class*="modal-footer">  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Hủy</button>  <form *method*="POST" *action*="{{ route('orders.destroy', $each) }}">  @csrf  @method('DELETE')  <button *type*="submit" *class*="btn btn-danger">Xóa</button>  </form>  </div>  </div>  </div>  </div>  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  @if ($data->lastPage() > 1)  <div *class*="card-footer">  <div *class*="paginate">  {{ $data->links('pagination::bootstrap-5')}}  </div>  </div>  @endif  </div>  </div>  </div>  </div>  @if(session('message') && session('type'))  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body bg-{{ session('type') }} d-flex align-items-center justify-content-between">  <div *class*="d-flex justify-content-center align-items-center gap-2">  @if(session('type') == 'success')  <i *class*="fas fa-check-circle text-light fs-5"></i>  @elseif(session('type') == 'danger' || session('type') == 'warning')  <i *class*="fas fa-xmark-circle text-light fs-5"></i>  @elseif(session('type') == 'info' || session('type') == 'secondary')  <i *class*="fas fa-info-circle text-light fs-5"></i>  @endif  <h6 *class*="h6 text-white m-0">{{ session('message') }}</h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endif  @endsection  @push('javascript')  <script *type*="text/javascript">  $(document).ready(function () {  const toastLiveExample = $('#liveToast');  const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));  toastBootstrap.show();  })  </script>  @endpush |
| --- |

* Trong file code trên: Sẽ có một giao diện hiển thị các đơn hàng trong hệ thống.

#### **Giao diện thêm đơn thùng hàng**

File: **resources\views\simples\create.blade.php**

****

| @extends('layouts.master')  @section('title', 'Tạo đơn thùng hàng')  @section('content')  <div *class*="container">  <div *class*="row pb-5">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-75">  <div *class*="card mb-2">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin chung</h4>  </div>  <div *class*="card-body">  <form *method*="POST" *id*="formInformation">  <input *type*="hidden" *name*="isSimple" *value*="1">  @csrf  <div *class*="row">  <div *class*="col-md-4">  <div *class*="row">  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_Id\_Customer"  *style*="width: 130px;">Khách hàng</label>  <select *class*="form-select selectValidate" *id*="FK\_Id\_Customer" *name*="FK\_Id\_Customer">  @foreach ($customers as $each)  <option *value*="{{ $each->Id\_Customer }}">{{ $each->Name\_Customer }}</option>  @endforeach  </select>  </div>  <span *class*="form-message text-danger"></span>  </div>  <div *class*="col-md-12">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 130px;">Ngày đặt hàng</label>  <input *type*="date" *class*="form-control" *id*="Date\_Order" *name*="Date\_Order"  *value*="{{ \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  </div>  </div>  <div *class*="col-md-4">  <div *class*="row">  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 135px;">  Ngày giao hàng  </label>  <input *type*="date" *class*="form-control" *id*="Date\_Dilivery" *name*="Date\_Dilivery"  *value*="{{ \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 135px;">  Ngày nhận hàng  </label>  <input *type*="date" *class*="form-control" *id*="Date\_Reception" *name*="Date\_Reception"  *value*="{{ \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group">  <span *class*="input-group-text bg-secondary-subtle">Ghi chú</span>  <textarea *class*="form-control" *style*="height: 91px;" *aria-label*="Notes" *name*="Note"  *rows*="5"></textarea>  </div>  </div>  </div>  </form>  </div>  </div>  <div *class*="card">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin thùng hàng</h4>  </div>  <div *class*="card-body">  <form *method*="POST" *id*="formProduct">  @csrf  <div *class*="row">  <div *class*="col-md-6">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_Id\_RawMaterial" *style*="width: 130px;">  Nguyên vật liệu  </label>  <select *name*="FK\_Id\_RawMaterial" *id*="FK\_Id\_RawMaterial" *class*="form-select">  @foreach ($materials as $each)  <option *value*="{{ $each->Id\_RawMaterial }}" *data-name*="{{ $each->Name\_RawMaterial }}">  {{ $each->Name\_RawMaterial }}  </option>  @endforeach  </select>  </div>  </div>  <div *class*="col-md-6">  <div *class*="input-group mb-3 align-items-center ">  <label *class*="input-group-text bg-secondary-subtle" *for*="Count\_RawMaterial">  Số lượng nguyên vật liệu  </label>  <input *type*="number" *name*="Count\_RawMaterial" *id*="Count\_RawMaterial" *class*="form-control" *min*="1"  *value*='1'>  <p *data-name*="unit" *class*="m-0 ps-3"></p>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_ID\_ContainerType" *style*="width: 130px;">  Thùng chứa  </label>  <select *class*="form-select selectValidate" *name*="FK\_ID\_ContainerType" *id*="FK\_ID\_ContainerType">  @foreach ($containers as $each)  <option *value*="{{ $each->Id\_ContainerType }}" *data-name*="{{ $each->Name\_ContainerType }}">  {{ $each->Name\_ContainerType }}  </option>  @endforeach  </select>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="Count\_Container">  Số lượng thùng chứa  </label>  <input *type*="number" *name*="Count\_Container" *id*="Count\_Container" *class*="form-control" *min*="1"  *value*='1'>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="Price\_Container">  Đơn giá  </label>  <input *type*="number" *name*="Price\_Container" *id*="Price\_Container" *class*="form-control" *min*="1"  *value*='1'>  </div>  </div>  </div>  <button *type*="submit" *class*="btn btn-primary-color mt-3 px-5">  <i *class*="fa-solid fa-plus text-white"></i>  Thêm sản phẩm  </button>  </form>  </div>  <div *class*="card-footer mt-3 p-0">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *style*="width: 150px;" *scope*="col">Nguyên liệu</th>  <th *class*="text-center" *scope*="col">Số lượng nguyên liệu</th>  <th *class*="text-center" *scope*="col">Đơn vị</th>  <th *class*="text-center" *scope*="col">Thùng chứa</th>  <th *class*="text-center" *scope*="col">Số lượng thùng chứa</th>  <th *class*="text-center" *scope*="col">Đơn giá</th>  <th></th>  </tr>  </thead>  <tbody *id*="table-data">  </tbody>  </table>  </div>  </div>  <div *class*="d-flex align-items-center justify-content-end mt-3">  <button *type*="submit" *class*="btn btn-lg btn-primary-color px-4" *id*="saveBtn">Lưu</button>  </div>  </div>  </div>  </div>  </div>  @endsection  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body d-flex align-items-center justify-content-between">  <div *class*="d-flex justify-content-center align-items-center gap-2">  <i *id*="icon" *class*="fas text-light fs-5"></i>  <h6 *id*="toast-msg" *class*="h6 text-white m-0"></h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @push('javascript')  <script *src*="{{ asset('js/simples/createSimple.js') }}"></script>  <script *src*="{{ asset('js/eventHandler.js') }}"></script>  @endpush |
| --- |

* Trong file code trên: Sẽ có 2 form, 1 form thêm đơn hàng, 1 form thêm thùng hàng. Khi ấn button “Thêm sản phẩm” thì sản phẩm sẽ được hiển thị dạng bảng ngay bên dưới (chưa gửi dữ liệu lên máy chủ). Khi ấn button “Lưu” thì sẽ thêm đơn hàng mới với những thông tin tại form thứ nhất và sẽ thêm những thùng hàng mà người dùng đã nhập tại bảng vào cơ sở dữ liệu (Sử dụng Ajax và JQuery để cập nhật dữ liệu).
* File Javascript: createSimple.js **(public\js\simple\createSimple.js)**

$(document).ready(function () {

const toastLiveExample = $("#liveToast");

const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));

let count = 0;

function validateDate(*firstControl*, *secondControl*) {

$(firstControl).on("change", function () {

let firstControlValue = **new** Date($(*this*).val());

let secondControlValue = **new** Date($(secondControl).val());

let formMessage = $(*this*).parent().next();

if (firstControlValue < secondControlValue) {

formMessage.html("Vui lòng chọn ngày phù hợp");

$(*this*).addClass("is-invalid");

} else {

formMessage.html("");

$(*this*).removeClass("is-invalid");

}

});

}

$("#formProduct").on("submit", function (*event*) {

event.preventDefault();

let token = $('meta[name="csrf-token"]').attr("content");

let form = $(*this*);

let url = "/simples/addSimple";

let unit = $("p[data-name='unit']").html();

$.ajax({

url: url,

type: "post",

data: {

unit: unit,

formData: form.serialize(),

\_token: token,

},

success: function (*response*) {

count++;

let htmls = "";

$.each(response.data, function (*key*, *value*) {

let rawMaterialId = value.FK\_Id\_RawMaterial;

let rawMaterialName = $(

`#FK\_Id\_RawMaterial option[value="${rawMaterialId}"]`

).data("name");

let containerTypeId = value.FK\_ID\_ContainerType;

let containerTypeName = $(

`#FK\_ID\_ContainerType option[value="${containerTypeId}"]`

).data("name");

htmls += `<tr data-id="${count}">

<td class="text-center" data-id="rawMaterialId" data-value="${rawMaterialId}">

${rawMaterialName}

</td>

<td class="text-center" data-id="Count\_RawMaterial" data-value="${value.Count\_RawMaterial}">

${value.Count\_RawMaterial}

</td>

<td class="text-center">${value.unit}</td>

<td class="text-center" data-id="containerTypeId" data-value="${containerTypeId}">

${containerTypeName}

</td>

<td class="text-center" data-id="Count\_Container" data-value="${value.Count\_Container}">

${value.Count\_Container}

</td>

<td class="text-center" data-id="Price\_Container" data-value="${value.Price\_Container}">

${value.formattedPrice}

</td>

<td>

<button type="button" class="btn btn-sm btn-outline-light text-primary-color border-secondary" data-bs-toggle="modal" data-bs-target="#deleteRow${count}">

<i class="fa-solid fa-trash"></i>

</button>

<div class="modal fade" id="deleteRow${count}" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="exampleModalLabel">Xác nhận</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

Bạn có chắc chắn về việc sản phẩm này

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Hủy</button>

<button type="button" class="btn btn-danger btnDelete" data-id="${count}">Xóa</button>

</div>

</div>

</div>

</div>

</td>

</tr>`;

});

$("#table-data").append(htmls);

*// Xóa dữ liệu đã nhập/chọn trong form*

form[0].reset();

$(".toast-body").addClass("bg-success");

$("#icon").addClass("fa-check-circle");

$("#toast-msg").html("Thêm thùng hàng thành công");

toastBootstrap.show();

},

});

});

$("#saveBtn").on("click", function () {

let token = $('meta[name="csrf-token"]').attr("content");

let form = $("#formInformation");

let formData = form.serialize();

let url = "/orders/storeOrder";

$.ajax({

url: url,

type: "post",

data: {

formData: formData,

\_token: token,

},

success: function (*response*) {

let secondUrl = "/simples/storeSimple";

let rowDataArray = [];

$("#table-data tr").each(function () {

let row = $(*this*);

let rowData = {};

rowData.FK\_Id\_RawMaterial = row

.find('td[data-id="rawMaterialId"]')

.data("value");

rowData.Count\_RawMaterial = row

.find('td[data-id="Count\_RawMaterial"]')

.data("value");

rowData.FK\_Id\_ContainerType = row

.find('td[data-id="containerTypeId"]')

.data("value");

rowData.Count\_Container = row

.find('td[data-id="Count\_Container"]')

.data("value");

rowData.Price\_Container = row

.find('td[data-id="Price\_Container"]')

.data("value");

rowData.FK\_Id\_Order = *response*.id;

rowDataArray.push(rowData);

});

$.ajax({

url: secondUrl,

type: "post",

data: {

rowData: rowDataArray,

\_token: token,

},

success: function (*response*) {

*// Điều hướng về trang chủ*

window.location.href = "/orders/";

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(*xhr*.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(*xhr*.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

});

$(document).on("click", ".btnDelete", function () {

let id = $(*this*).data("id");

let rowElement = $(*this*).closest('tr[data-id="' + id + '"]');

let modalElement = $("#deleteRow" + id); *// Lấy modal tương ứng với hàng*

modalElement.on("hidden.bs.modal", function () {

*// Xóa hàng khi modal được ẩn*

rowElement.remove();

});

$(".toast-body").addClass("bg-success");

$("#icon").addClass("fa-check-circle");

$("#toast-msg").html("Xóa thùng hàng thành công");

toastBootstrap.show();

*// Đóng modal*

modalElement.modal("hide");

});

let dateOrderControl = $("input[name='Date\_Order']");

let deliveryDateControl = $("input[name='Date\_Dilivery']");

let receptionDateControl = $("input[name='Date\_Reception']");

dateOrderControl.on("change", function () {

let selectedDate = **new** *Date*($(*this*).val());

let currentDate = **new** *Date*();

let formMessage = $(*this*).parent().next();

if (selectedDate > currentDate) {

formMessage.html("Vui lòng chọn ngày phù hợp");

$(*this*).addClass("is-invalid");

} else {

formMessage.html("");

$(*this*).removeClass("is-invalid");

}

});

validateDate(deliveryDateControl, dateOrderControl);

validateDate(receptionDateControl, dateOrderControl);

validateDate(receptionDateControl, deliveryDateControl);

});

* Trong file JavaScript trên:
  + Hàm ValidateDate() để kiểm tra các ô: Ngày đặt hàng, Ngày giao hàng và ngày nhận hàng. Nếu ô nào không hợp lệ thì sẽ báo ra cho người dùng.
  + Sự kiện formProduct submit là sự kiện sẽ đẩy thông tin thùng hàng mà người dùng nhập vào table, sau đó reset hết các ô input để người dùng nhập tiếp.
  + Sự kiện saveBtn là sự kiện thêm form “Thông tin chung” vào máy chủ, sau đó cập nhật từng dòng của bảng vào máy chủ.
  + Sự kiện btnDelete là sự kiện xóa từng hàng dữ liệu mà người dùng vừa nhập.
* File Javascript: eventHandler.js **(public\js\eventHandler.js)**

$(document).ready(function () {

let selectElement = $('select[name="FK\_Id\_RawMaterial"]');

selectElement.on("change", function () {

let token = $('meta[name="csrf-token"]').attr("content");

let id = $(*this*).val();

$.ajax({

url: "/rawMaterials/showUnit",

method: "POST",

dataType: "json",

data: {

id: id,

\_token: token,

},

success: function (*data*) {

let unitElement = $("[data-name='unit']");

unitElement.html(*data*.unit);

},

});

});

let firstOptionValue = $(selectElement).val();

*// Gán giá trị cho phần tử select*

$(selectElement).val(firstOptionValue);

*// Gọi sự kiện change để hiển thị dữ liệu*

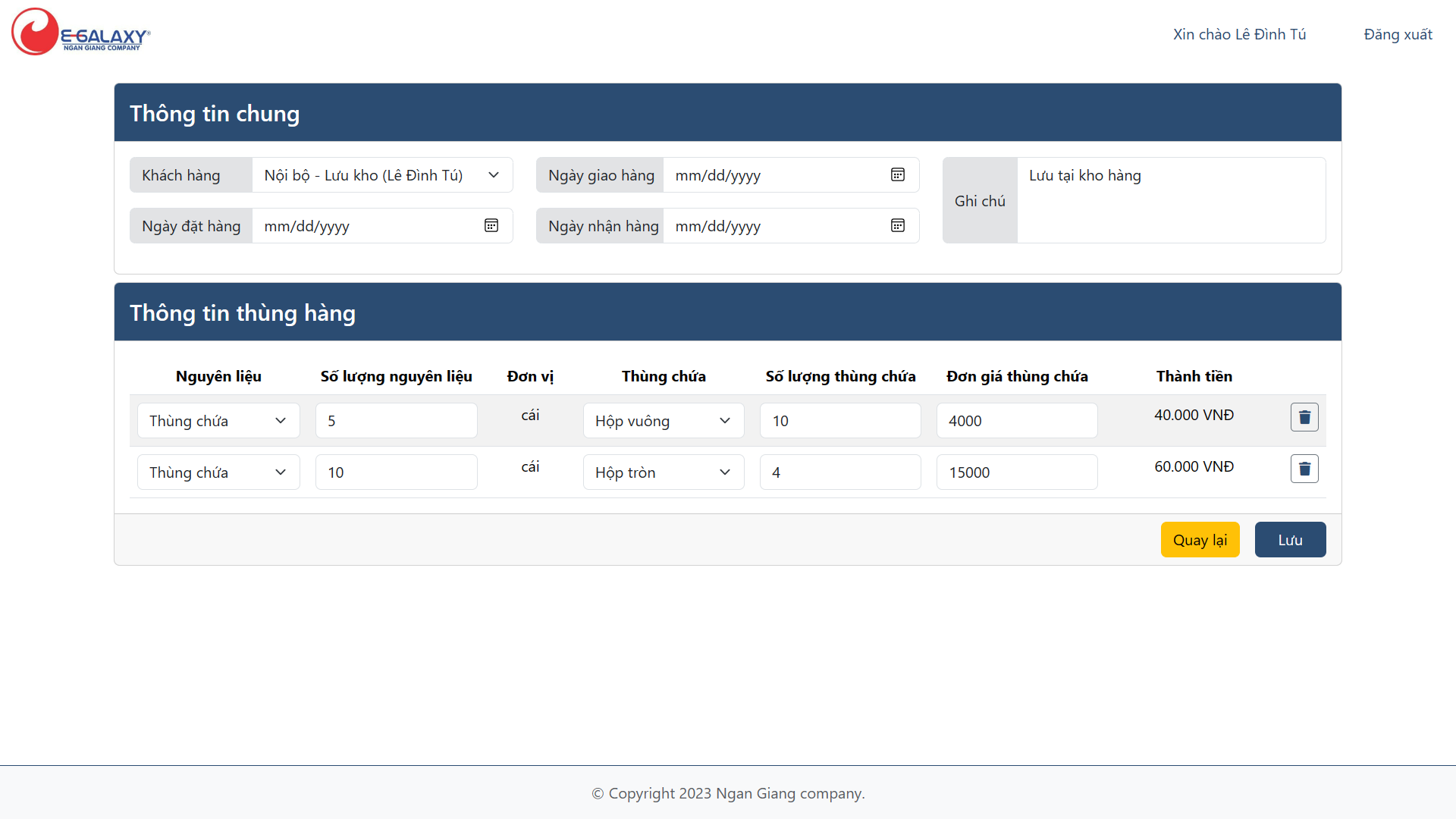
$(selectElement).change();

});

* Trong file JavaScript trên:
  + Sự kiện selectElement change là sự kiện sẽ hiển thị được đơn vị tương ứng với nguyên liệu mà người dùng chọn bằng Ajax.

#### **Giao diện sửa đơn thùng hàng**

File: **resources\views\simples\edit.blade.php**

****

| @extends('layouts.master')  @section('title', 'Sửa đơn thùng hàng')  @section('content')  <div *class*="container">  <div *class*="row pb-5">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-100">  <div *class*="card mb-2">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin chung</h4>  </div>  <div *class*="card-body">  <form *method*="POST" *id*="formInformation">  @csrf  <input *type*="hidden" *name*="Id\_Order" *value*="{{ $order->Id\_Order }}">  <div *class*="row">  <div *class*="col-md-4">  <div *class*="row">  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_Id\_Customer"  *style*="width: 130px;">Khách hàng</label>  <select *class*="form-select selectValidate" *id*="FK\_Id\_Customer" *name*="FK\_Id\_Customer">  @foreach($customers as $each)  @if($each->Id\_Customer == $order->FK\_Id\_Customer)  <option *value*="{{ $each->Id\_Customer }}" *selected*>{{ $each->Name\_Customer }}</option>  @else  <option *value*="{{ $each->Id\_Customer }}">{{ $each->Name\_Customer }}</option>  @endif  @endforeach  </select>  </div>  <span *class*="form-message text-danger"></span>  </div>  <div *class*="col-md-12">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 130px;">Ngày đặt hàng</label>  <input *type*="date" *class*="form-control" *id*="Date\_Order" *name*="Date\_Order"  *value*="{{ $order->order\_date }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  </div>  </div>  <div *class*="col-md-4">  <div *class*="row">  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 135px;">  Ngày giao hàng  </label>  <input *type*="date" *class*="form-control" *id*="Date\_Dilivery" *name*="Date\_Dilivery"  *value*="{{ $order->delivery\_date }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 135px;">  Ngày nhận hàng  </label>  <input *type*="date" *class*="form-control" *id*="Date\_Reception" *name*="Date\_Reception"  *value*="{{ $order->reception\_date }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group">  <span *class*="input-group-text bg-secondary-subtle">Ghi chú</span>  <textarea *class*="form-control" *style*="height: 91px;" *aria-label*="Notes" *name*="Note"  *rows*="5">{{ $order->Note}}</textarea>  </div>  </div>  </div>  </form>  </div>  </div>  <div *class*="card">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin thùng hàng</h4>  </div>  <div *class*="card-body">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *class*="text-center" *scope*="col" *width*="200">Nguyên liệu</th>  <th *class*="text-center" *scope*="col" *width*="200">Số lượng nguyên liệu</th>  <th *class*="text-center" *scope*="col" *width*="100">Đơn vị</th>  <th *class*="text-center" *scope*="col" *width*="200">Thùng chứa</th>  <th *class*="text-center" *scope*="col" *width*="200">Số lượng thùng chứa</th>  <th *class*="text-center" *scope*="col" *width*="200">Đơn giá thùng chứa</th>  <th *class*="text-center" *scope*="col" *width*="200">Thành tiền</th>  <th *scope*="col"></th>  </tr>  </thead>  <tbody *id*="table-data">  @foreach($data as $each)  <tr *class*="js-row" *data-id*="{{ $each->Id\_SimpleContent }}">  <td>  <select *class*="form-select selectValidate" *id*="FK\_Id\_RawMaterial" *name*="FK\_Id\_RawMaterial">  @foreach($materials as $material)  @if($material->Id\_RawMaterial == $each->FK\_Id\_RawMaterial)  <option *value*="{{ $material->Id\_RawMaterial }}" *selected*>{{ $material->Name\_RawMaterial }}  </option>  @else  <option *value*="{{ $material->Id\_RawMaterial }}">{{ $material->Name\_RawMaterial }}</option>  @endif  @endforeach  </select>  </td>  <td *class*="text-center">  <input *class*="form-control" *type*="number" *name*="Count\_RawMaterial" *id*="Count\_RawMaterial"  *value*="{{ $each->Count\_RawMaterial}}" *min*="0">  </td>  <td *class*="text-center" *data-name*="unit">{{ $each->unit}}</td>  <td *class*="text-center">  <select *class*="form-select selectValidate" *id*="FK\_Id\_ContainerType" *name*="FK\_Id\_ContainerType">  @foreach($containers as $container)  @if($container->Id\_ContainerType == $each->FK\_Id\_ContainerType)  <option *value*="{{ $container->Id\_ContainerType }}" *selected*>{{ $container->Name\_ContainerType }}  </option>  @else  <option *value*="{{ $container->Id\_ContainerType }}">{{ $container->Name\_ContainerType }}</option>  @endif  @endforeach  </select>  </td>  <td *class*="text-center">  <input *class*="form-control" *type*="number" *name*="Count\_Container" *id*="Count\_Container"  *value*="{{ $each->Count\_Container}}" *min*="0">  </td>  <td *class*="text-center">  <input *class*="form-control" *type*="number" *name*="Price\_Container" *id*="Price\_Container"  *value*="{{ number\_format($each->Price\_Container, 0, '', '')}}" *min*="0">  </td>  <td *class*="text-center" *data-id*="total">  {{ number\_format($each->Price\_Container \* $each->Count\_Container, 0, ',', '.'). ' VNĐ' }}  </td>  <td *class*="text-center">  <button *type*="button" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary"  *data-bs-toggle*="modal" *data-bs-target*="#deleteID-{{$each->Id\_SimpleContent}}">  <i *class*="fa-solid fa-trash"></i>  </button>  <div *class*="modal fade" *id*="deleteID-{{$each->Id\_SimpleContent}}" *tabindex*="-1"  *aria-labelledby*="exampleModalLabel" *aria-hidden*="true">  <div *class*="modal-dialog">  <div *class*="modal-content">  <div *class*="modal-header">  <h1 *class*="modal-title fs-5" *id*="exampleModalLabel">Xác nhận</h1>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  Bạn có chắc chắn về việc sản phẩm này  </div>  <div *class*="modal-footer">  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Hủy</button>  <button *type*="button" *class*="btn btn-danger btnDelete"  *data-id*="{{$each->Id\_SimpleContent}}">Xóa</button>  </div>  </div>  </div>  </div>  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  <div *class*="card-footer d-flex align-items-center justify-content-end gap-3">  <a *href*="{{ route('orders.index') }}" *class*="btn btn-warning">Quay lại</a>  <button *type*="submit" *class*="btn btn-primary-color px-4" *id*="saveBtn">Lưu</button>  </div>  </div>  </div>  </div>  </div>  </div>  @endsection  @push('javascript')  <script *src*="{{ asset('js/simples/editSimple.js') }}"></script>  <script *src*="{{ asset('js/eventHandler.js') }}"></script>  @endpush |
| --- |

* Trong file code trên: Sẽ có 2 form, 1 form thêm đơn hàng, 1 form thêm thùng hàng. Khi ấn button “Thêm sản phẩm” thì sản phẩm sẽ được hiển thị dạng bảng ngay bên dưới (chưa gửi dữ liệu lên máy chủ). Khi ấn button “Lưu” thì sẽ thêm đơn hàng mới với những thông tin tại form thứ nhất và sẽ thêm những thùng hàng mà người dùng đã nhập tại bảng vào cơ sở dữ liệu (Sử dụng Ajax và JQuery để cập nhật dữ liệu).
* File Javascript: editSimple.js **(public\js\simples\editSimple.js)**

$(document).ready(function () {

$("input[name='Count\_Container']").on("change", function () {

let rowElement = $(*this*).closest(".js-row");

let price = rowElement.find("#Price\_Container").val();

let totalPrice = parseInt(price) \* parseInt($(*this*).val());

var formattedPrice = totalPrice.toLocaleString("vi-VN") + " VNĐ";

rowElement.find("[data-id='total']").html(formattedPrice);

});

$("input[name='Price\_Container']").on("change", function () {

let rowElement = $(*this*).closest(".js-row");

let count = rowElement.find("#Count\_Container").val();

let totalPrice = parseInt(count) \* parseInt($(*this*).val());

var formattedPrice = totalPrice.toLocaleString("vi-VN") + " VNĐ";

rowElement.find("[data-id='total']").html(formattedPrice);

});

$("#saveBtn").on("click", function () {

let token = $('meta[name="csrf-token"]').attr("content");

let form = $("#formInformation");

let formData = form.serialize();

let url = "/orders/updateOrder";

let id = $('input[name="Id\_Order"').val();

$.ajax({

url: url,

type: "post",

data: {

formData: formData,

\_token: token,

id: id,

},

success: function (*response*) {

let secondUrl = "/simples/updateSimple";

let rowDataArray = [];

$(".js-row").each(function () {

let row = $(*this*);

let rowData = {};

rowData.Id\_SimpleContent = row.data("id");

rowData.FK\_Id\_RawMaterial = row

.find("#FK\_Id\_RawMaterial")

.val();

rowData.Count\_RawMaterial = row

.find("#Count\_RawMaterial")

.val();

rowData.FK\_Id\_ContainerType = row

.find("#FK\_Id\_ContainerType")

.val();

rowData.Count\_Container = row

.find("#Count\_Container")

.val();

rowData.Price\_Container = row

.find("#Price\_Container")

.val();

rowDataArray.push(rowData);

});

$.ajax({

url: secondUrl,

type: "post",

data: {

rowData: rowDataArray,

\_token: token,

},

success: function (*response*) {

*// Điều hướng về trang chủ*

window.location.href = "/orders/";

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

});

$(document).on("click", ".btnDelete", function () {

let id = $(*this*).data("id");

let modalElement = $("#deleteID-" + id); *// Lấy modal tương ứng với hàng*

let token = $('meta[name="csrf-token"]').attr("content");

let rowElement = $(*this*).closest('tr[data-id="' + id + '"]');

*// Xóa hàng khi modal được ẩn*

$.ajax({

url: "/simples/deleteSimple",

method: "POST",

dataType: "json",

data: {

id: id,

\_token: token,

},

success: function (*data*) {

modalElement.on("hidden.bs.modal", function () {

rowElement.remove();

});

*// Đóng modal*

modalElement.modal("hide");

},

});

});

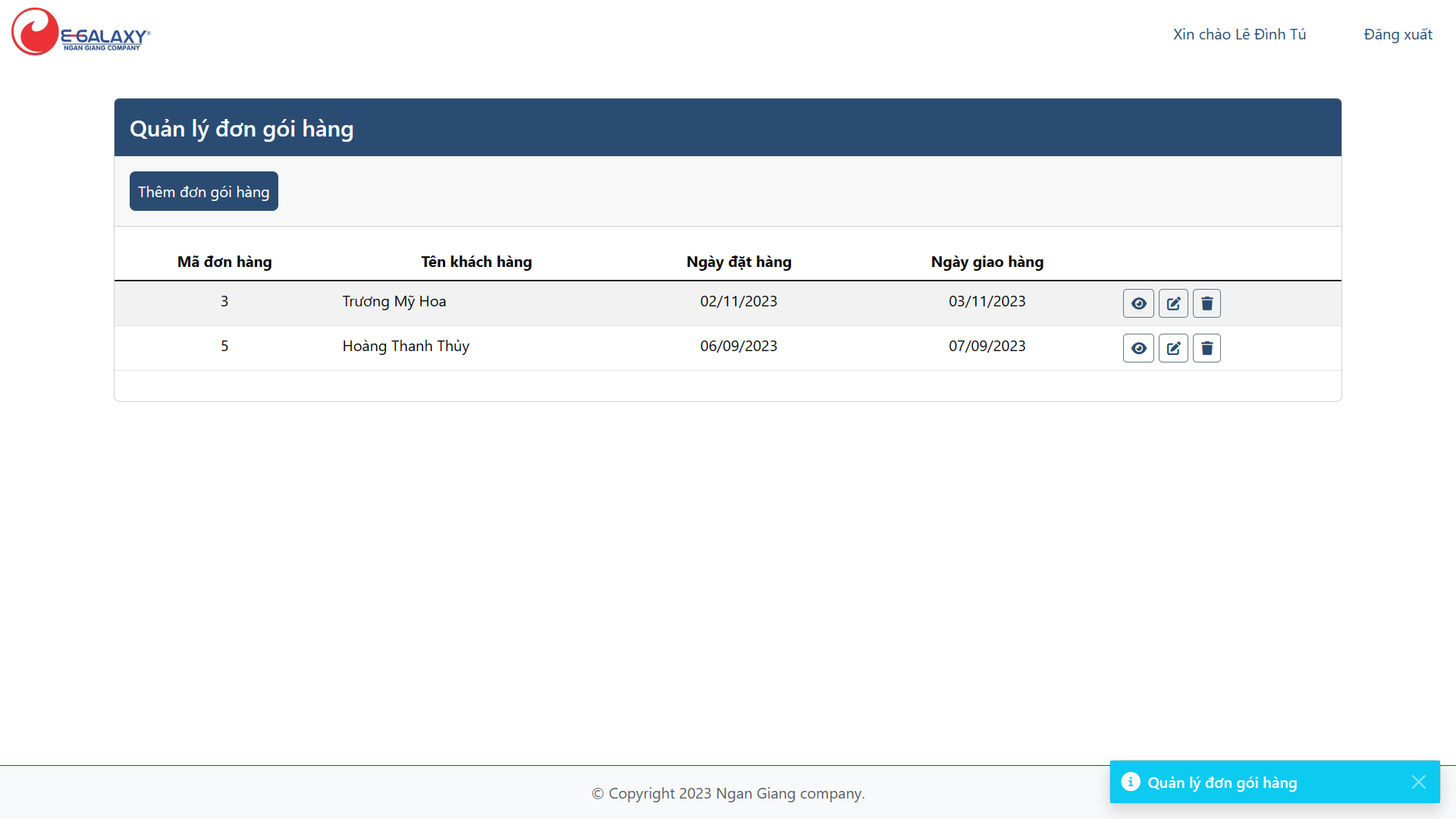
});

* Trong file JavaScript trên:
  + Sự kiện [Count\_Container, change] là khi mà người dùng thay đổi số lượng thùng hàng thì sẽ cập nhật “Thành tiền”
  + Sự kiện [Price\_Container, change] là khi mà người dùng thay đổi đơn giá thùng hàng thì sẽ cập nhật “Thành tiền”
  + Sự kiện [saveBtn, click] là khi mà người dùng click button “Lưu” thì sẽ lưu lại tất cả dữ liệu người dùng đã thay đổi vào cơ sở dữ liệu và điều hướng về trang Quản lý đơn hàng
  + Sự kiện [btnDelete, click] là khi mà người dùng muốn xóa đơn hàng, thì sẽ xóa luôn trong máy chủ và xóa luôn dòng đó tại bảng.

### **Giao diện quản lý đơn gói hàng**

#### **Giao diện hiển thị đơn gói hàng**

File: **resources\views\packs\index.blade.php**

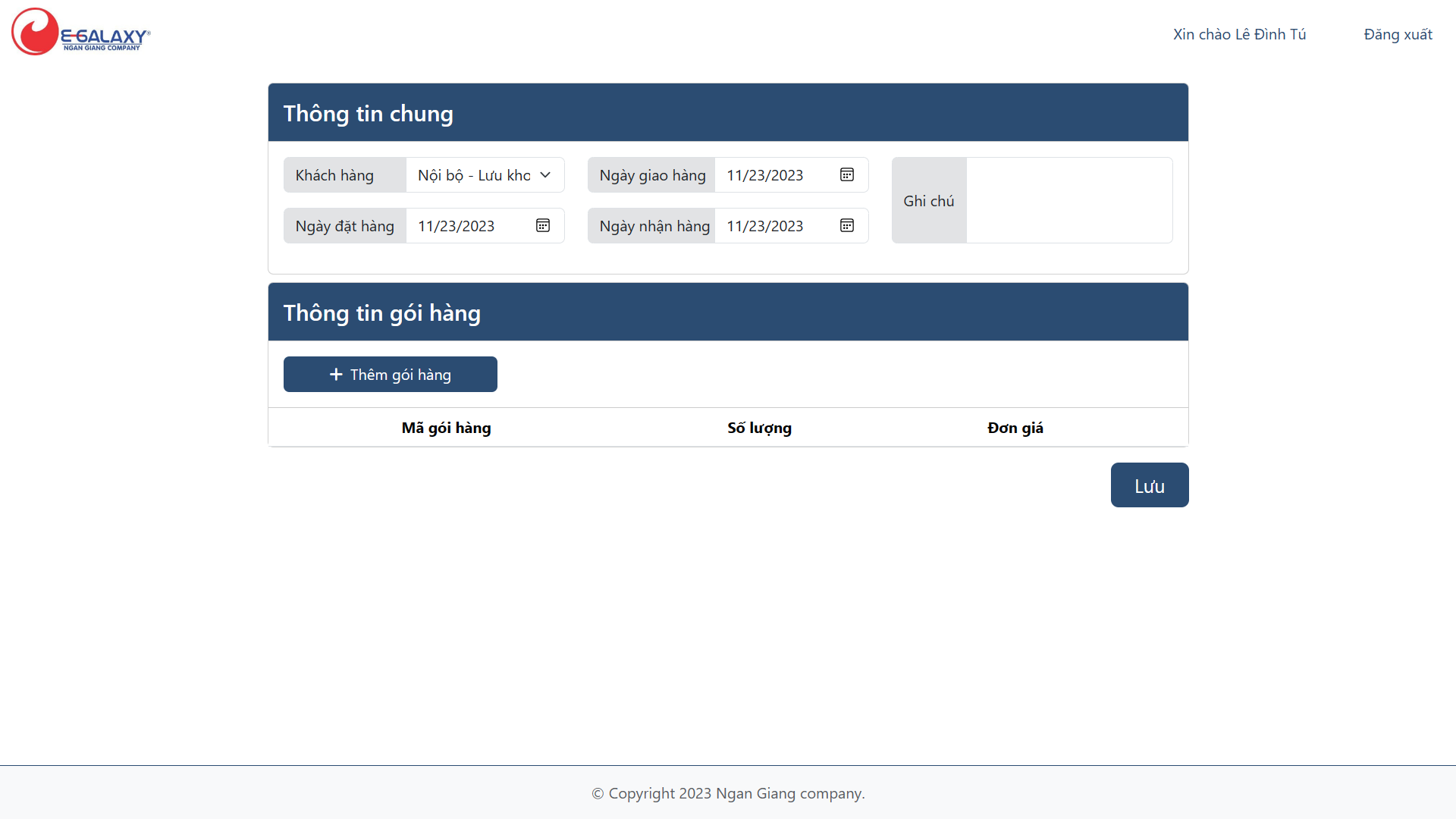


| @extends('layouts.master')  @section('title', 'Quản lý đơn gói hàng')  @section('content')  <div *class*="container">  <div *class*="row">  <div *class*="col-md-12">  <div *class*="card mt-3">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Quản lý đơn gói hàng</h4>  <a *href*="{{ route('packs.create')}}" *class*="btn btn-primary-color text-white p-2 my-3 ms-3">  Thêm đơn gói hàng  </a>  </div>  <div *class*="card-body px-0">  <table *class*="table table-striped w-100">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Mã đơn hàng</th>  <th *class*="text-center" *scope*="col">Tên khách hàng</th>  <th *class*="text-center" *scope*="col">Ngày đặt hàng</th>  <th *class*="text-center" *scope*="col">Ngày giao hàng</th>  </tr>  </thead>  <tbody *id*="table-data" *class*="table-group-divider">  @foreach($data as $each)  <tr>  <td *class*="text-center">{{ $each->Id\_Order }}</td>  <td>{{ $each->customer->Name\_Customer}}</td>  <td *class*="text-center">{{ $each->order\_date}}</td>  <td *class*="text-center">{{ $each->delivery\_date}}</td>  <td>  <a *href*="" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-eye"></i>  </a>  <a *href*="" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-pen-to-square"></i>  </a>  <button *type*="button" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary"  *data-bs-toggle*="modal" *data-bs-target*="#deleteOrder-{{$each->Id\_Order}}">  <i *class*="fa-solid fa-trash"></i>  </button>  <div *class*="modal fade" *id*="deleteOrder-{{$each->Id\_Order}}" *tabindex*="-1"  *aria-labelledby*="exampleModalLabel" *aria-hidden*="true">  <div *class*="modal-dialog">  <div *class*="modal-content">  <div *class*="modal-header">  <h1 *class*="modal-title fs-5" *id*="exampleModalLabel">Xác nhận</h1>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  Bạn có chắc chắn về việc sản phẩm này  </div>  <div *class*="modal-footer">  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Hủy</button>  <form *method*="POST" *action*="">  @csrf  @method('DELETE')  <button *type*="submit" *class*="btn btn-danger">Xóa</button>  </form>  </div>  </div>  </div>  </div>  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  @if ($data->lastPage() > 1)  <div *class*="card-footer">  <div *class*="paginate">  {{ $data->links('pagination::bootstrap-5')}}  </div>  </div>  @endif  </div>  </div>  </div>  </div>  @if(session('message') && session('type'))  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body bg-{{ session('type') }} d-flex align-items-center justify-content-between">  <div *class*="d-flex justify-content-center align-items-center gap-2">  @if(session('type') == 'success')  <i *class*="fas fa-check-circle text-light fs-5"></i>  @elseif(session('type') == 'danger' || session('type') == 'warning')  <i *class*="fas fa-xmark-circle text-light fs-5"></i>  @elseif(session('type') == 'info' || session('type') == 'secondary')  <i *class*="fas fa-info-circle text-light fs-5"></i>  @endif  <h6 *class*="h6 text-white m-0">{{ session('message') }}</h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endif  @endsection  @push('javascript')  <script *type*="text/javascript">  $(document).ready(function () {  const toastLiveExample = $('#liveToast');  const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));  toastBootstrap.show();  })  </script>  @endpush |
| --- |

* Trong file code trên: Sẽ có một giao diện hiển thị các đơn gói hàng trong hệ thống.

#### **Giao diện thêm đơn gói hàng**

File: **resources\views\packs\create.blade.php**

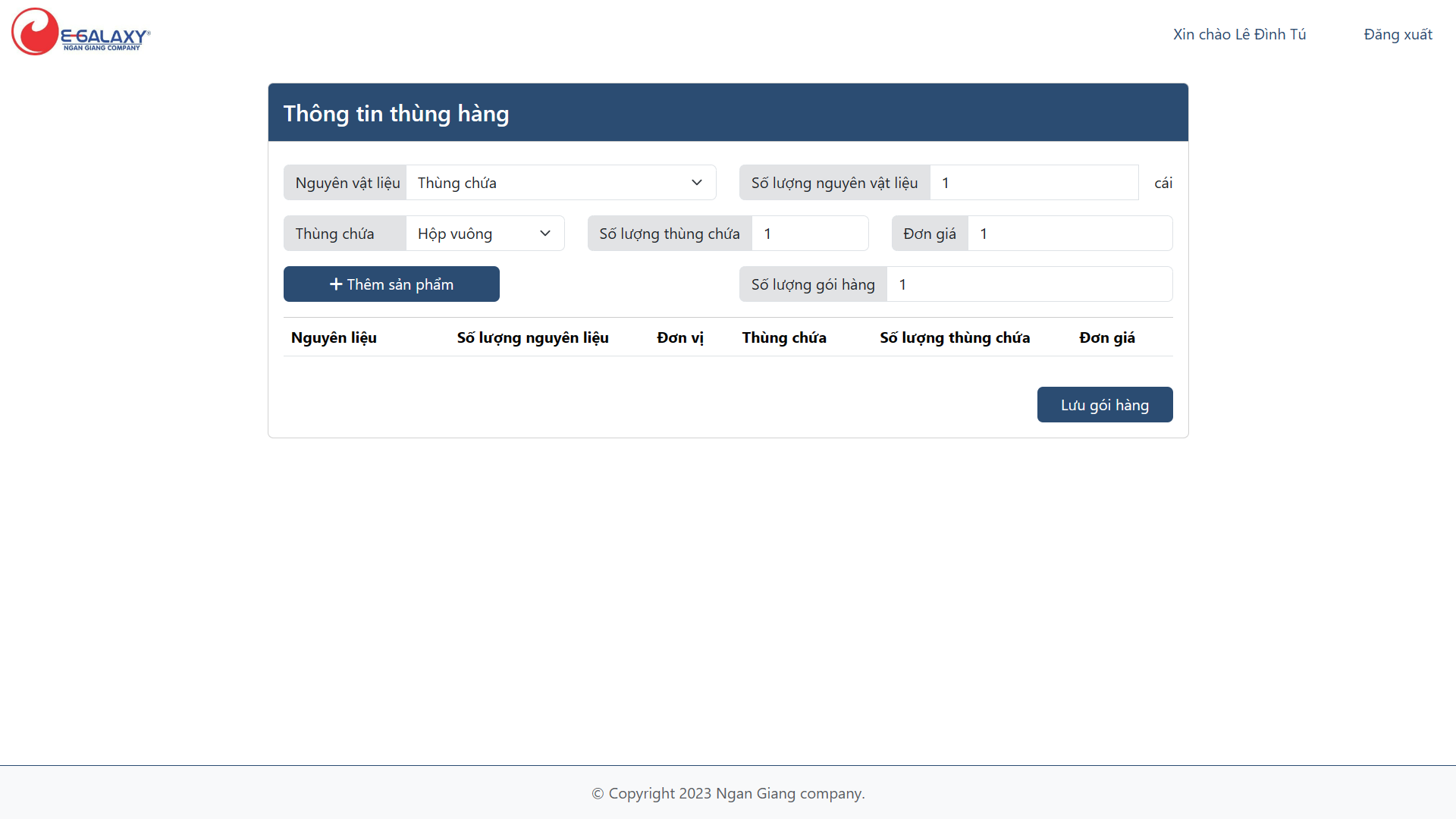
****

| @extends('layouts.master')  @section('title', 'Tạo gói hàng')  @section('content')  <div *class*="container">  <div *class*="row pb-5">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-75">  <div *class*="card mb-2">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin chung</h4>  </div>  <div *class*="card-body">  <form *method*="POST" *id*="formInformation">  @csrf  <input *type*="hidden" *name*="count" *value*="{{ isset($count) ? $count : 0}}">  <input *type*="hidden" *name*="isSimple" *value*="0">  <div *class*="row">  <div *class*="col-md-4">  <div *class*="row">  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_Id\_Customer"  *style*="width: 130px;">Khách hàng</label>  <select *class*="form-select selectValidate" *id*="FK\_Id\_Customer" *name*="FK\_Id\_Customer">  @foreach ($customers as $each)  @if (isset($information))  @if ($information->FK\_Id\_Customer == $each->Id\_Customer)  <option *value*="{{ $each->Id\_Customer }}" *selected*>{{ $each->Name\_Customer }}</option>  @else  <option *value*="{{ $each->Id\_Customer }}">{{ $each->Name\_Customer }}</option>  @endif  @else  <option *value*="{{ $each->Id\_Customer }}">{{ $each->Name\_Customer }}</option>  @endif  @endforeach  </select>  </div>  <span *class*="form-message text-danger"></span>  </div>  <div *class*="col-md-12">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 130px;">Ngày đặt hàng</label>  <input *type*="date" *class*="form-control" *id*="Date\_Order" *name*="Date\_Order"  *value*="{{ isset($information) ? \Carbon\Carbon::parse($information->Date\_Order)->format('Y-m-d') : \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  </div>  </div>  <div *class*="col-md-4">  <div *class*="row">  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 135px;">  Ngày giao hàng  </label>  <input *type*="date" *class*="form-control" *id*="Date\_Dilivery" *name*="Date\_Dilivery"  *value*="{{ isset($information) ? \Carbon\Carbon::parse($information->Date\_Dilivery)->format('Y-m-d') : \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  <div *class*="col-md-12 mb-3">  <div *class*="input-group">  <label *class*="input-group-text bg-secondary-subtle" *style*="width: 135px;">  Ngày nhận hàng  </label>  <input *type*="date" *class*="form-control" *id*="Date\_Reception" *name*="Date\_Reception"  *value*="{{ isset($information) ? \Carbon\Carbon::parse($information->Date\_Reception)->format('Y-m-d') : \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  <span *class*="form-message text-danger"></span>  </div>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group">  <span *class*="input-group-text bg-secondary-subtle">Ghi chú</span>  <textarea *class*="form-control" *style*="height: 91px;" *aria-label*="Notes" *name*="Note"  *rows*="5">{{ isset($information) ? $information->Note : '' }}</textarea>  </div>  </div>  </div>  </form>  </div>  </div>  <div *class*="card">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin gói hàng</h4>  </div>  <div *class*="card-body">  <button *type*="submit" *class*="btn btn-primary-color px-5" *id*="addBtn">  <i *class*="fa-solid fa-plus text-white me-2"></i>Thêm gói hàng  </button>  </div>  <div *class*="card-footer p-0">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Mã gói hàng</th>  <th *class*="text-center" *scope*="col">Số lượng</th>  <th *class*="text-center" *scope*="col">Đơn giá</th>  <th></th>  </tr>  </thead>  <tbody *id*="table-data">  @if (isset($data))  @foreach ($data as $each)  <tr *data-id*="{{ $each->Id\_PackContent }}">  <td *class*="text-center">{{ $each->Id\_PackContent }}</td>  <td *class*="text-center">{{ $each->Count\_Pack }} gói hàng</td>  <td *class*="text-center">{{ number\_format($each->Price\_Pack, 0, ',', '.') . ' VNĐ' }}</td>  <td *class*="text-center">  <button *type*="button" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary"  *data-bs-toggle*="modal" *data-bs-target*="#deleteID-{{ $each->Id\_PackContent }}">  <i *class*="fa-solid fa-trash"></i>  </button>  <div *class*="modal fade" *id*="deleteID-{{ $each->Id\_PackContent }}" *tabindex*="-1"  *aria-labelledby*="exampleModalLabel" *aria-hidden*="true">  <div *class*="modal-dialog">  <div *class*="modal-content">  <div *class*="modal-header">  <h1 *class*="modal-title fs-5" *id*="exampleModalLabel">Xác nhận</h1>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  Bạn có chắc chắn về việc sản phẩm này  </div>  <div *class*="modal-footer">  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Hủy</button>  <button *type*="button" *class*="btn btn-danger deletePack"  *data-id*="{{ $each->Id\_PackContent }}">Xóa</button>  </div>  </div>  </div>  </div>  </td>  </tr>  @endforeach  @endif  </tbody>  </table>  </div>  </div>  <div *class*="d-flex align-items-center justify-content-end mt-3">  <a *href*="{{ route('packs.index') }}" *class*="btn btn-lg btn-primary-color px-4" *id*="btn\_save">Lưu</a>  </div>  </div>  </div>  </div>  </div>  @endsection  @push('javascript')  <script *src*="{{ asset('js/packs/createPack.js') }}"></script>  @endpush |
| --- |

* Trong file code trên: Sẽ có 2 form, 1 form thêm đơn hàng, 1 form thêm gói hàng.Lần đầu người dùng nhập button “Thêm gói hàng” thì sẽ insert thông tin chung người dùng đã nhập vào database, sau đó chuyển đến form “'Thêm thùng hàng vào gói hàng'”, còn những lần sau thì sẽ điều hướng luôn chứ không có thêm dữ liệu người dùng đã nhập ở thông tin chung vào database.

#### **Giao diện thêm gói hàng vào lô hàng**

File: **resources\views\packs\createPack.blade.php**

****

| @extends('layouts.master')  @section('title', 'Thêm thùng hàng vào gói hàng')  @section('content')  <div *class*="container">  <div *class*="row pb-5">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-75">  <div *class*="card">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Thông tin thùng hàng</h4>  </div>  <h1>{{ session('count') }}</h1>  <div *class*="card-body">  <input *type*="hidden" *name*="FK\_Id\_Order" *value*="{{ $\_GET['id'] }}">  <form *method*="POST" *id*="formProduct">  @csrf  <div *class*="row">  <div *class*="col-md-6">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_Id\_RawMaterial" *style*="width: 130px;">  Nguyên vật liệu  </label>  <select *name*="FK\_Id\_RawMaterial" *id*="FK\_Id\_RawMaterial" *class*="form-select">  @foreach($materials as $each)  <option *value*="{{ $each->Id\_RawMaterial }}" *data-name*="{{ $each->Name\_RawMaterial }}">  {{ $each->Name\_RawMaterial }}  </option>  @endforeach  </select>  </div>  </div>  <div *class*="col-md-6">  <div *class*="input-group mb-3 align-items-center ">  <label *class*="input-group-text bg-secondary-subtle" *for*="Count\_RawMaterial">  Số lượng nguyên vật liệu  </label>  <input *type*="number" *name*="Count\_RawMaterial" *id*="Count\_RawMaterial" *class*="form-control" *min*="1"  *value*='1'>  <p *data-name*="unit" *class*="m-0 ps-3"></p>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="FK\_ID\_ContainerType" *style*="width: 130px;">  Thùng chứa  </label>  <select *class*="form-select selectValidate" *name*="FK\_ID\_ContainerType" *id*="FK\_ID\_ContainerType">  @foreach($containers as $each)  <option *value*="{{ $each->Id\_ContainerType }}" *data-name*="{{ $each->Name\_ContainerType }}">  {{ $each->Name\_ContainerType }}  </option>  @endforeach  </select>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="Count\_Container">  Số lượng thùng chứa  </label>  <input *type*="number" *name*="Count\_Container" *id*="Count\_Container" *class*="form-control" *min*="1"  *value*='1'>  </div>  </div>  <div *class*="col-md-4">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="Price\_Container">  Đơn giá  </label>  <input *type*="number" *name*="Price\_Container" *id*="Price\_Container" *class*="form-control" *min*="1"  *value*='1'>  </div>  </div>  <div *class*="col-md-6">  <button *type*="submit" *class*="btn btn-primary-color mb-3 px-5">  <i *class*="fa-solid fa-plus text-white"></i>  Thêm sản phẩm  </button>  </div>  <div *class*="col-md-6">  <div *class*="input-group mb-3">  <label *class*="input-group-text bg-secondary-subtle" *for*="Count\_Pack">  Số lượng gói hàng  </label>  <input *type*="number" *name*="Count\_Pack" *id*="Count\_Pack" *class*="form-control" *min*="1" *value*='1'>  </div>  </div>  </form>  </div>  <div *class*="card-footer p-0">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *style*="width: 150px;" *scope*="col">Nguyên liệu</th>  <th *class*="text-center" *scope*="col">Số lượng nguyên liệu</th>  <th *class*="text-center" *scope*="col">Đơn vị</th>  <th *class*="text-center" *scope*="col">Thùng chứa</th>  <th *class*="text-center" *scope*="col">Số lượng thùng chứa</th>  <th *class*="text-center" *scope*="col">Đơn giá</th>  <th></th>  </tr>  </thead>  <tbody *id*="table-data">  </tbody>  </table>  </div>  </div>  <div *class*="d-flex align-items-center justify-content-end my-3 me-3">  <button *type*="submit" *class*="btn btn-primary-color px-4" *id*="saveBtn">Lưu gói hàng</button>  </div>  </div>  </div>  </div>  </div>  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body d-flex align-items-center justify-content-between">  <div *class*="d-flex justify-content-center align-items-center gap-2">  <i *id*="icon" *class*="fas text-light fs-5"></i>  <h6 *id*="toast-msg" *class*="h6 text-white m-0"></h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endsection  @push('javascript')  <script *src*="{{ asset('js/packs/createPack.js') }}"></script>  <script *src*="{{ asset('js/eventHandler.js') }}"></script>  @endpush |
| --- |

* Trong file code trên sẽ có một form thêm thùng hàng vào Gói hàng, sau khi lưu thì nó sẽ trả về form thêm Gói hàng (form có thông tin chung và nút chuyển form thêm chi tiết gói hàng)
* File Javascript: createPack.js **(public\js\packs\createPack.js)**

$(document).ready(function () {

const toastLiveExample = $("#liveToast");

const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));

let count = $("input[name='count']").val();

$("#addBtn").on("click", function () {

if (count == 0) {

let token = $('meta[name="csrf-token"]').attr("content");

let form = $("#formInformation");

let formData = form.serialize();

let url = "/orders/storeOrder";

count++;

$.ajax({

url: url,

type: "post",

data: {

formData: formData,

\_token: token,

},

success: function (*response*) {

console.log(response);

window.location.href =

"/packs/createPack?id=" + response.id;

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

} else if (count == 1) {

let urlParams = **new** URLSearchParams(window.location.search);

let id = urlParams.get("id");

window.location.href = "/packs/createPack?id=" + id;

}

});

$("#formProduct").on("submit", function (*event*) {

*event*.preventDefault();

let token = $('meta[name="csrf-token"]').attr("content");

let form = $(*this*);

let url = "/simples/addSimple";

let unit = $("p[data-name='unit']").html();

let count = 0;

$.ajax({

url: url,

type: "post",

data: {

unit: unit,

formData: form.serialize(),

\_token: token,

},

success: function (*response*) {

count++;

let htmls = "";

$.each(response.data, function (*key*, *value*) {

let rawMaterialId = value.FK\_Id\_RawMaterial;

let rawMaterialName = $(

`#FK\_Id\_RawMaterial option[value="${rawMaterialId}"]`

).data("name");

let containerTypeId = value.FK\_ID\_ContainerType;

let containerTypeName = $(

`#FK\_ID\_ContainerType option[value="${containerTypeId}"]`

).data("name");

htmls += `<tr data-id="${count}">

<td class="text-center" data-id="rawMaterialId" data-value="${rawMaterialId}">

${rawMaterialName}

</td>

<td class="text-center" data-id="Count\_RawMaterial" data-value="${value.Count\_RawMaterial}">

${value.Count\_RawMaterial}

</td>

<td class="text-center">${value.unit}</td>

<td class="text-center" data-id="containerTypeId" data-value="${containerTypeId}">

${containerTypeName}

</td>

<td class="text-center" data-id="Count\_Container" data-value="${value.Count\_Container}">

${value.Count\_Container}

</td>

<td class="text-center" data-id="Price\_Container" data-value="${value.Price\_Container}">

${value.formattedPrice}

</td>

<td>

<button type="button" class="btn btn-sm btn-outline-light text-primary-color border-secondary" data-bs-toggle="modal" data-bs-target="#deleteRow${count}">

<i class="fa-solid fa-trash"></i>

</button>

<div class="modal fade" id="deleteRow${count}" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="exampleModalLabel">Xác nhận</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

Bạn có chắc chắn về việc sản phẩm này

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Hủy</button>

<button type="button" class="btn btn-danger btnDelete" data-id="${count}">Xóa</button>

</div>

</div>

</div>

</div>

</td>

</tr>`;

});

$("#table-data").append(htmls);

*// Xóa dữ liệu đã nhập/chọn trong form*

form[0].reset();

$("#icon").addClass("fa-check-circle");

$(".toast-body").addClass("bg-success");

$("#toast-msg").html("Thêm thùng hàng thành công");

toastBootstrap.show();

},

});

});

$("#saveBtn").on("click", function () {

let token = $('meta[name="csrf-token"]').attr("content");

let url = "/simples/storeSimple";

let order\_Id = $("input[name='FK\_Id\_Order']").val();

let rowDataArray = [];

let Price\_Pack = 0;

$("#table-data tr").each(function () {

let row = $(*this*);

let rowData = {};

rowData.FK\_Id\_RawMaterial = row

.find('td[data-id="rawMaterialId"]')

.data("value");

rowData.Count\_RawMaterial = row

.find('td[data-id="Count\_RawMaterial"]')

.data("value");

rowData.FK\_Id\_ContainerType = row

.find('td[data-id="containerTypeId"]')

.data("value");

rowData.Count\_Container = row

.find('td[data-id="Count\_Container"]')

.data("value");

rowData.Price\_Container = row

.find('td[data-id="Price\_Container"]')

.data("value");

rowData.FK\_Id\_Order = order\_Id;

rowDataArray.push(rowData);

let subtotal = rowData.Count\_Container \* rowData.Price\_Container; *// Tính subtotal*

Price\_Pack += subtotal;

});

$.ajax({

url: url,

type: "post",

data: {

rowData: rowDataArray,

\_token: token,

},

success: function (*response*) {

let countPack = $("input[name='Count\_Pack'");

let countValue = countPack.val();

let packData = {

Count\_Pack: countValue,

Price\_Pack: Price\_Pack,

FK\_Id\_Order: order\_Id,

};

let secondUrl = "/packs/storePack";

$.ajax({

url: secondUrl,

type: "post",

data: {

packData: packData,

\_token: token,

},

success: function (*response*) {

let thirdUrl = "/packs/storeDetail";

let Id\_PackContent = response.id;

$.ajax({

url: thirdUrl,

type: "post",

data: {

FK\_Id\_Order: order\_Id,

Id\_PackContent: Id\_PackContent,

\_token: token,

},

success: function (*response*) {

window.location.href =

"/packs/create?id=" + response.id;

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

});

$(document).on("click", ".btnDelete", function () {

let id = $(*this*).data("id");

let rowElement = $(*this*).closest('tr[data-id="' + id + '"]');

let modalElement = $("#deleteRow" + id); *// Lấy modal tương ứng với hàng*

modalElement.on("hidden.bs.modal", function () {

*// Xóa hàng khi modal được ẩn*

rowElement.remove();

});

*// Đóng modal*

$("#icon").addClass("fa-check-circle");

$(".toast-body").addClass("bg-success");

$("#toast-msg").html("Xóa thùng hàng thành công");

toastBootstrap.show();

modalElement.modal("hide");

});

$(".deletePack").each(function (*index*, *element*) {

$(*this*).on("click", function () {

let id = $(*this*).data("id");

let urlParams = **new** URLSearchParams(window.location.search);

let Id\_Order = urlParams.get("id");

let modalElement = $("#deleteID-" + id); *// Lấy modal tương ứng với hàng*

let token = $('meta[name="csrf-token"]').attr("content");

let rowElement = $(*this*).closest('tr[data-id="' + id + '"]');

*// Xóa hàng khi modal được ẩn*

$.ajax({

url: "/packs/deletePack",

method: "POST",

dataType: "json",

data: {

id: id,

Id\_Order: Id\_Order,

\_token: token,

},

success: function (*data*) {

modalElement.on("hidden.bs.modal", function () {

rowElement.remove();

});

*// Đóng modal*

modalElement.modal("hide");

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

});

});

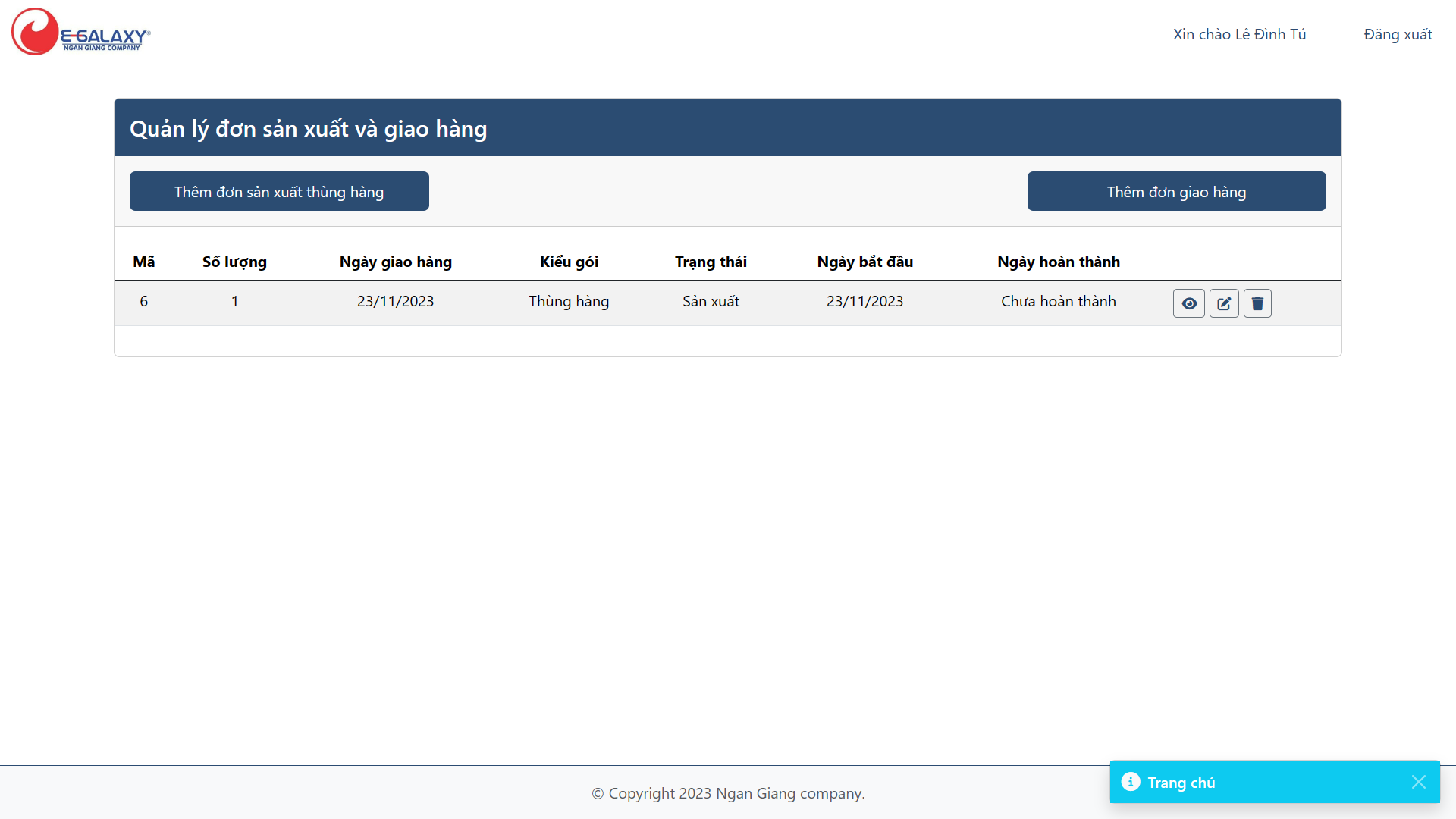
});

* Trong file JavaScript trên:
  + Sự kiện addBtn (trong form “Thêm gói hàng”): Sẽ kiểm tra biến count, nếu count = 0 thì sẽ thêm thông tin đơn hàng vào database rồi điều hướng đến form “Thêm thùng hàng vào gói hàng”, còn nếu count khác 0 thì sẽ chỉ điều hướng mà không them dữ liệu vào database.
  + Sự kiện saveBtn (trong form “Thêm thùng hàng vào gói hàng”): Tại Ajax lần đầu tiên, sẽ duyệt tất cả các table row và lấy dữ liệu từ đó để thêm vào database, sau đó tại Ajax lần thứ 2 thì sẽ thêm dữ liệu vào bảng ContentPack, và Ajax lần thứ 3 sẽ thêm dữ liệu vào bảng trung gian DetailContentSimpleOfPack.
  + Sự kiện deletePack (trong form “Thêm gói hàng”): Sẽ xóa dữ liệu tại 3 bảng: ContentSimple, ContentPack và DetailContentSimpleOfPack tương ứng với gói hàng mà người dùng muốn xóa.

### **Giao diện quản lý đơn sản xuất và giao hàng**

#### **Giao diện hiển thị đơn sản xuất và giao hàng**

File: **resources\views\makesOrExpeditions\index.blade.php**

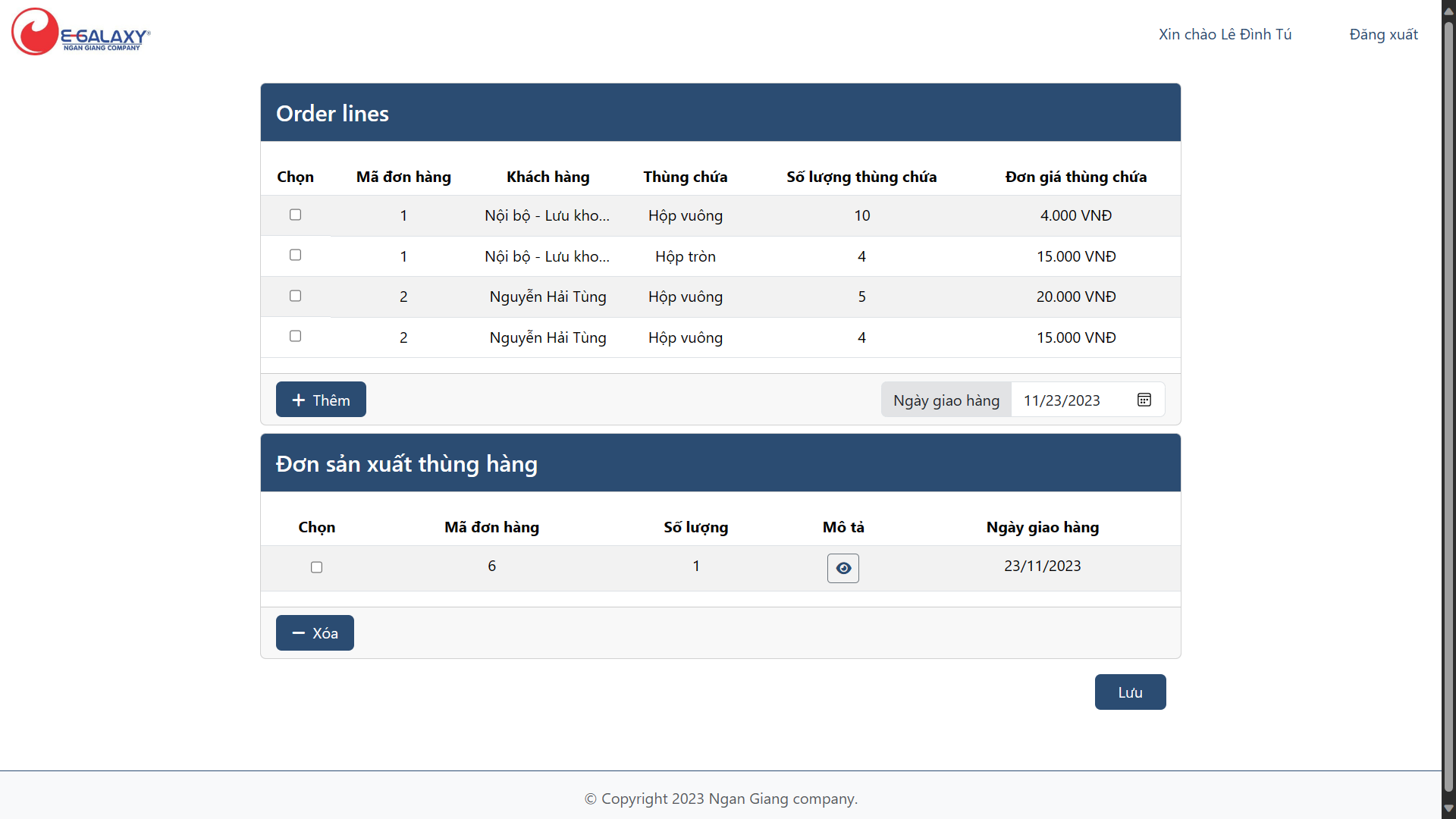


| @extends('layouts.master')  @section('title', 'Quản lý đơn sản xuất và giao hàng')  @section('content')  <div *class*="container">  <div *class*="row">  <div *class*="col-md-12">  <div *class*="card mt-3">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Quản lý đơn sản xuất và giao hàng</h4>  <div *class*="d-flex align-items-center justify-content-between my-3 px-3">  <a *href*="{{ route('makesOrExpeditions.create')}}" *class*="btn btn-primary-color text-white p-2 w-25">  Thêm đơn sản xuất thùng hàng  </a>  <a *href*="{{ route('orders.create')}}" *class*="btn btn-primary-color text-white p-2 w-25">  Thêm đơn giao hàng  </a>  </div>  </div>  <div *class*="card-body px-0">  <table *class*="table table-striped w-100">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Mã</th>  <th *class*="text-center" *scope*="col">Số lượng</th>  <th *class*="text-center" *scope*="col">Ngày giao hàng</th>  <th *class*="text-center" *scope*="col">Kiểu gói</th>  <th *class*="text-center" *scope*="col">Trạng thái</th>  <th *class*="text-center" *scope*="col">Ngày bắt đầu</th>  <th *class*="text-center" *scope*="col">Ngày hoàn thành</th>  <th></th>  </tr>  </thead>  <tbody *id*="table-data" *class*="table-group-divider">  @foreach($data as $each)  <tr>  <td *class*="text-center">{{ $each->Id\_OrderMakeOrExpedition }}</td>  <td *class*="text-center">{{ $each->Count }}</td>  <td *class*="text-center">{{ $each->delivery\_date }}</td>  <td *class*="text-center">{{ $each->type }}</td>  <td *class*="text-center">{{ $each->status }}</td>  <td *class*="text-center">{{ $each->start\_date }}</td>  <td *class*="text-center">{{ $each->finally\_date }}</td>  <td>  <a *href*="{{ route('orders.show', $each) }}"  *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-eye"></i>  </a>  <a *href*="{{ route('orders.edit', $each) }}"  *class*="btn btn-sm btn-outline-light text-primary-color border-secondary">  <i *class*="fa-solid fa-pen-to-square"></i>  </a>  <button *type*="button" *class*="btn btn-sm btn-outline-light text-primary-color border-secondary"  *data-bs-toggle*="modal" *data-bs-target*="#deleteOrder-{{$each->Id\_OrderMakeOrExpedition}}">  <i *class*="fa-solid fa-trash"></i>  </button>  <div *class*="modal fade" *id*="deleteOrder-{{$each->Id\_OrderMakeOrExpedition}}" *tabindex*="-1"  *aria-labelledby*="exampleModalLabel" *aria-hidden*="true">  <div *class*="modal-dialog">  <div *class*="modal-content">  <div *class*="modal-header">  <h1 *class*="modal-title fs-5" *id*="exampleModalLabel">Xác nhận</h1>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  Bạn có chắc chắn về việc sản phẩm này  </div>  <div *class*="modal-footer">  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Hủy</button>  <form *method*="POST" *action*="{{ route('orders.destroy', $each) }}">  @csrf  @method('DELETE')  <button *type*="submit" *class*="btn btn-danger">Xóa</button>  </form>  </div>  </div>  </div>  </div>  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  @if ($data->lastPage() > 1)  <div *class*="card-footer">  <div *class*="paginate">  {{ $data->links('pagination::bootstrap-5')}}  </div>  </div>  @endif  </div>  </div>  </div>  </div>  @if(session('message') && session('type'))  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body bg-{{ session('type') }} d-flex align-items-center justify-content-between">  <div *class*="d-flex justify-content-center align-items-center gap-2">  @if(session('type') == 'success')  <i *class*="fas fa-check-circle text-light fs-5"></i>  @elseif(session('type') == 'danger' || session('type') == 'warning')  <i *class*="fas fa-xmark-circle text-light fs-5"></i>  @elseif(session('type') == 'info' || session('type') == 'secondary')  <i *class*="fas fa-info-circle text-light fs-5"></i>  @endif  <h6 *class*="h6 text-white m-0">{{ session('message') }}</h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endif  @endsection  @push('javascript')  <script *type*="text/javascript">  $(document).ready(function () {  const toastLiveExample = $('#liveToast');  const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));  toastBootstrap.show();  })  </script>  @endpush |
| --- |

* Trong file code trên: Sẽ có một giao diện hiển thị các đơn sản xuất và gioa hàng trong hệ thống

#### **Giao diện thêm đơn sản xuất và giao hàng**

File: **resources\views\makesOrExpeditions\simples\create.blade.php**

****

| @extends('layouts.master')  @section('title', 'Tạo đơn sản xuất thùng hàng')  @section('content')  <div *class*="container">  <div *class*="row pb-5">  <div *class*="col-md-12 d-flex justify-content-center">  <div *class*="w-75">  <div *class*="card mb-2">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Order lines</h4>  </div>  <div *class*="card-body px-0">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Chọn</th>  <th *class*="text-center" *scope*="col">Mã đơn hàng</th>  <th *class*="text-center" *scope*="col">Khách hàng</th>  <th *class*="text-center" *scope*="col">Thùng chứa</th>  <th *class*="text-center" *scope*="col">Số lượng thùng chứa</th>  <th *class*="text-center" *scope*="col">Đơn giá thùng chứa</th>  </tr>  </thead>  <tbody *id*="table-data">  @foreach($data as $each)  <tr>  <td *class*="d-flex justify-content-center" *data-id*="Id\_SimpleContent"  *data-value*="{{ $each->Id\_SimpleContent }}">  <input *type*="checkbox" *class*="form-check" *name*="firstFormCheck"  *id*="firstFormCheck-{{ $each->Id\_SimpleContent }}">  </td>  <td *data-id*="Id\_Order" *data-value*="{{ $each->Id\_Order }}" *class*="text-center">{{ $each->Id\_Order }}  </td>  <td *data-id*="Name\_Customer" *data-value*="{{ $each->Name\_Customer }}" *class*="text-center text-truncate"  *style*="max-width: 150px; width: 150px;">  {{ $each->Name\_Customer }}  </td>  <td *data-id*="Name\_ContainerType" *data-value*="{{ $each->Name\_ContainerType }}" *class*="text-center">{{  $each->Name\_ContainerType }}  </td>  <td *data-id*="Count\_Container" *data-value*="{{ $each->Count\_Container }}" *class*="text-center">{{  $each->Count\_Container }}</td>  <td *data-id*="Price\_Container" *data-value*="{{ $each->Price\_Container }}" *class*="text-center">  {{ number\_format($each->Price\_Container, 0, ',', '.') }} VNĐ  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  <div *class*="card-footer">  <div *class*="d-flex align-content-center justify-content-between">  <button *type*="submit" *class*="btn btn-primary-color px-3" *id*="addBtn">  <i *class*="fa-solid fa-plus text-white me-2"></i>Thêm  </button>  <div *class*="input-group" *style*="width: 300px;">  <label *class*="input-group-text bg-secondary-subtle" *for*="DateDilivery">  Ngày giao hàng  </label>  <input *type*="date" *name*="DateDilivery" *id*="DateDilivery" *class*="form-control"  *value*="{{ \Carbon\Carbon::now()->format('Y-m-d') }}">  </div>  </div>  </div>  </div>  <div *class*="card mb-2">  <div *class*="card-header p-0 overflow-hidden">  <h4 *class*="card-title m-0 bg-primary-color p-3">Đơn sản xuất thùng hàng</h4>  </div>  <div *class*="card-body px-0">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Chọn</th>  <th *class*="text-center" *scope*="col">Mã đơn hàng</th>  <th *class*="text-center" *scope*="col">Số lượng</th>  <th *class*="text-center" *scope*="col">Mô tả</th>  <th *class*="text-center" *scope*="col">Ngày giao hàng</th>  </tr>  </thead>  <tbody *id*="table-result">  @foreach($orders as $each)  <tr>  <td *class*="align-middle">  <div *class*="d-flex justify-content-center align-items-center">  <input *type*="checkbox" *class*="form-check" *name*="secondFormCheck"  *id*="secondFormCheck-{{ $each->Id\_OrderMakeOrExpedition}}">  </div>  </td>  <td *data-id*="Id\_OrderMakeOrExpedition" *data-value*="{{ $each->Id\_OrderMakeOrExpedition }}"  *class*="text-center">  {{ $each->Id\_OrderMakeOrExpedition }}  </td>  <td *class*="text-center">  {{ $each->Count }}  </td>  <td *class*="text-center">  <button *type*="button" *class*="btnShow btn btn-sm btn-outline-light  text-primary-color border-secondary" *data-bs-toggle*="modal"  *data-bs-target*="#show-{{ $each->Id\_OrderMakeOrExpedition }}"  *data-id*="{{ $each->Id\_OrderMakeOrExpedition }}">  <i *class*="fa-solid fa-eye"></i>  </button>  <div *class*="modal fade" *id*="show-{{ $each->Id\_OrderMakeOrExpedition }}" *tabindex*="-1"  *aria-labelledby*="#show-{{ $each->Id\_OrderMakeOrExpedition }}Label" *aria-hidden*="true">  <div *class*="modal-dialog modal-lg">  <div *class*="modal-content">  <div *class*="modal-header p-2 bg-primary-color text-start" *data-bs-theme*="dark">  <h5 *class*="modal-title w-100 " *id*="show-{{ $each->Id\_OrderMakeOrExpedition }}Label">  Thông tin chi tiết đơn hàng số {{ $each->Id\_OrderMakeOrExpedition }}  </h5>  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>  </div>  <div *class*="modal-body">  <table *class*="table table-striped m-0">  <thead>  <tr>  <th *class*="text-center" *scope*="col">Nguyên liệu</th>  <th *class*="text-center" *scope*="col">Số lượng nguyên liệu</th>  <th *class*="text-center" *scope*="col">Đơn vị</th>  <th *class*="text-center" *scope*="col">Thùng chứa</th>  <th *class*="text-center" *scope*="col">Số lượng thùng chứa</th>  <th *class*="text-center" *scope*="col">Đơn giá</th>  </tr>  </thead>  <tbody *class*="table-simples" *class*="p-5"  *data-value*="{{ $each->Id\_OrderMakeOrExpedition}}">  </tbody>  </table>  </div>  </div>  </div>  </div>  </td>  <td *class*="text-center">  {{ \Carbon\Carbon::parse($each->DateDilivery)->format('d/m/Y') }}  </td>  </tr>  @endforeach  </tbody>  </table>  @if ($orders->lastPage() > 1)  <div *class*="card-footer">  <div *class*="paginate">  {{ $orders->links('pagination::bootstrap-5')}}  </div>  </div>  @endif  </div>  <div *class*="card-footer">  <button *type*="submit" *class*="btn btn-primary-color px-3" *id*="deleteBtn">  <i *class*="fa-solid fa-minus text-white me-2"></i>Xóa  </button>  </div>  </div>  <div *class*="d-flex align-items-center justify-content-end my-3 me-3">  <a *href*="{{ route('makesOrExpeditions.index')}}" *class*="btn btn-primary-color px-4">Lưu</a>  </div>  </div>  </div>  </div>  </div>  <div *class*="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div *id*="liveToast" *class*="toast" *role*="alert" *aria-live*="assertive" *aria-atomic*="true">  <div *class*="toast-body d-flex align-items-center justify-content-between">  <div *class*="d-flex justify-content-center align-items-center gap-2">  <i *id*="icon" *class*="fas text-light fs-5"></i>  <h6 *id*="toast-msg" *class*="h6 text-white m-0"></h6>  </div>  <button *type*="button" *class*="btn-close btn-close-white" *data-bs-dismiss*="toast" *aria-label*="Close"></button>  </div>  </div>  </div>  @endsection  @push('javascript')  <script *src*="{{ asset('js/makesOrExpeditions/simples/create.js') }}"></script>  <script *src*="{{ asset('js/eventHandler.js') }}"></script>  @endpush |
| --- |

* Trong file code trên: Sẽ có 2 bảng: Bảng đầu tiên là bảng hiển thị những thùng hàng mà chưa bắt đầu sản xuất, người dùng có thể chọn và thêm các thùng hàng vào khâu sản xuất. Tại bảng thứ 2, sẽ hiển thị những đơn sản xuất thùng hàng, người dùng có thể xem chi tiết (xem các thùng hàng của đơn) và có thể xóa đơn.
* File JavaScript: create.js (**public\js\makesOrExpeditions\simples\create.js)**

$(document).ready(function () {

let token = $('meta[name="csrf-token"]').attr("content");

const toastLiveExample = $("#liveToast");

const toastBootstrap = **new** bootstrap.Toast(toastLiveExample.get(0));

$("#addBtn").on("click", function () {

let rowElements = $("#table-data tr");

let rowDataArray = [];

let dateDeliveryInput = $("input[name='DateDilivery']");

let dateDeliveryInputValue = dateDeliveryInput.val();

let isValid = false;

rowElements.each(function () {

if ($(*this*).find("input[type=checkbox]").prop("checked") === true) {

let rowData = {};

rowData.Id\_SimpleContent = $(*this*)

.find('td[data-id="Id\_SimpleContent"]')

.data("value");

rowDataArray.push(rowData);

isValid = true;

}

});

if (isValid) {

let url = "/makesOrExpeditions/store";

$.ajax({

url: url,

type: "post",

data: {

rowData: rowDataArray,

dateValue: dateDeliveryInputValue,

\_token: token,

},

success: function (*response*) {

*// Xử lý dữ liệu thành công*

let newData = response.data;

let Id\_OrderMakeOrExpedition = response.id;

let Count = newData.Count;

let DateDilivery = newData.DateDilivery;

*// Tạo chuỗi HTML cho dữ liệu mới*

let html = `

<tr>

<td class="align-middle">

<div class="d-flex justify-content-center align-items-center">

<input type="checkbox" class="form-check" name="secondFormCheck" id="secondFormCheck-${Id\_OrderMakeOrExpedition}">

</div>

</td>

<td class="text-center">${Id\_OrderMakeOrExpedition}</td>

<td class="text-center">${Count}</td>

<td class="text-center">

<button type="button" class="btnShow btn btn-sm btn-outline-light

text-primary-color border-secondary" data-bs-toggle="modal"

data-bs-target="#show-${Id\_OrderMakeOrExpedition}"

data-id="${Id\_OrderMakeOrExpedition}">

<i class="fa-solid fa-eye"></i>

</button>

<div class="modal fade" id="show-${Id\_OrderMakeOrExpedition}" tabindex="-1"

aria-labelledby="show-${Id\_OrderMakeOrExpedition}Label" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header p-2 bg-primary-color text-start" data-bs-theme="dark">

<h5 class="modal-title w-100 " id="exampleModalLabel">

Thông tin chi tiết đơn hàng số ${Id\_OrderMakeOrExpedition}

</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<table class="table table-striped m-0">

<thead>

<tr>

<th class="text-center" scope="col">Nguyên liệu</th>

<th class="text-center" scope="col">Số lượng nguyên liệu</th>

<th class="text-center" scope="col">Đơn vị</th>

<th class="text-center" scope="col">Thùng chứa</th>

<th class="text-center" scope="col">Số lượng thùng chứa</th>

<th class="text-center" scope="col">Đơn giá</th>

</tr>

</thead>

<tbody class="table-simples" class="p-5" data-value="${Id\_OrderMakeOrExpedition}">

</tbody>

</table>

</div>

</div>

</div>

</div>

</td>

</td>

<td class="text-center">

${formatDate(DateDilivery)}

</td>

</tr>`;

*// Hàm chuyển đổi định dạng ngày tháng thành "dd/mm/yyyy"*

function formatDate(*date*) {

const d = **new** Date(date);

const day = String(d.getDate()).padStart(2, "0");

const month = String(d.getMonth() + 1).padStart(2, "0");

const year = d.getFullYear();

return `${day}/${month}/${year}`;

}

*// Thêm dữ liệu mới vào đầu bảng 'table-result'*

$("#table-result").prepend(html);

*// Bind click event to the view button for the newly added row*

bindViewButtonClick(

$("#table-result").find(".btnShow").first()

);

*// Các tác vụ khác sau khi hiển thị thành công*

$("#icon").addClass("fa-check-circle");

$(".toast-body").addClass("bg-success");

$("#toast-msg").html("Thêm đơn sản xuất thành công");

toastBootstrap.show();

rowElements.each(function () {

rowElements.each(function () {

$(*this*)

.find("input[type=checkbox]")

.prop("checked", false);

});

});

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

} else {

$(".toast-body").addClass("bg-warning");

$("#icon").addClass("fa-xmark-circle");

$("#toast-msg").html("Vui lòng chọn đơn thùng hàng");

toastBootstrap.show();

}

});

$("#deleteBtn").on("click", function () {

let rowElements = $("#table-result tr");

let rowDataArray = [];

let isValid = false;

rowElements.each(function () {

if ($(*this*).find("input[type=checkbox]").prop("checked") === true) {

let rowData = {};

rowData.Id\_OrderMakeOrExpedition = $(*this*)

.find('td[data-id="Id\_OrderMakeOrExpedition"]')

.data("value");

rowDataArray.push(rowData);

isValid = true;

}

});

if (isValid) {

let url = "/makesOrExpeditions/destroy";

$.ajax({

url: url,

type: "post",

data: {

rowData: rowDataArray,

\_token: token,

},

success: function (*response*) {

rowElements.each(function () {

if (

$(*this*)

.find("input[type=checkbox]")

.prop("checked") === true

) {

$(*this*).remove();

}

$(".toast-body").addClass("bg-success");

$("#icon").addClass("fa-check-circle");

$("#toast-msg").html("Xóa đơn sản xuất thành công");

toastBootstrap.show();

});

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(*xhr*.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

} else {

$(".toast-body").addClass("bg-warning");

$("#icon").addClass("fa-xmark-circle");

$("#toast-msg").html("Vui lòng chọn đơn sản xuất");

toastBootstrap.show();

}

});

bindViewButtonClick($(".btnShow"));

function bindViewButtonClick(*element*) {

*element*.on("click", function () {

let id = $(*this*).data("id");

$.ajax({

url: "/makesOrExpeditions/show",

method: "POST",

dataType: "json",

data: {

id: id,

\_token: token,

},

success: function (*response*) {

let table;

$(".table-simples").each(function () {

if ($(*this*).data("value") == id) {

table = $(*this*);

}

});

function numberFormat(

*number*,

*decimals*,

*decimalSeparator*,

*thousandSeparator*

) {

*decimals* = *decimals* !== undefined ? *decimals* : 0;

*decimalSeparator* = *decimalSeparator* || ",";

*thousandSeparator* = *thousandSeparator* || ".";

var parsedNumber = parseFloat(*number*);

var parts = parsedNumber.toFixed(*decimals*).split(".");

parts[0] = parts[0].replace(

/\B(?=(\d{3})+(?!\d))/g,

*thousandSeparator*

);

return parts.join(*decimalSeparator*);

}

let htmls = "";

$.each(response.data, function (*key*, *value*) {

htmls += `<tr>

<td class="text-center">${value.Name\_RawMaterial}</td>

<td class="text-center">${value.Count\_RawMaterial}</td>

<td class="text-center">${value.Name\_ContainerType}</td>

<td class="text-center">${value.Count\_Container}</td>

<td class="text-center">${value.Count\_Container}</td>

<td class="text-center">${numberFormat(

value.Price\_Container

)} VNĐ </td>

</tr>`;

});

table.html(htmls);

},

error: function (*xhr*) {

*// Xử lý lỗi khi gửi yêu cầu Ajax*

console.log(xhr.responseText);

alert("Có lỗi xảy ra. Vui lòng thử lại sau.");

},

});

});

}

});

* Trong file JavaScript trên:
  + Sự kiện [AddBtn, click] là khi mà người dùng chọn thùng hàng và nhấn nút “Thêm” thì tất cả các thùng hàng sẽ được thêm vào khâu sản xuất và được hiển thị ở bảng “Hiển thị đơn sản xuất thùng hàng”.
  + Sự kiện [deleteBtn, click] là khi mà người dùng chọn đơn sản xuất và nhấn nút “Xóa” thì tất cả các đơn người dùng chọn sẽ bị xóa trong máy chủ và bị xóa tại giao diện.

## **Bước 7:** Viết các hàm tương tích trong các file Controller

#### **HomeController**

File: **app/Http/Controllers/HomeController.php**

| class HomeController extends *Controller*  {  public function index()  {  if (!*Session*::has("type") && !*Session*::has("message")) {  *Session*::flash('type', 'info');  *Session*::flash('message', 'Trang chủ');  }  return view('index', ['productionStationLines' => *ProductionStationLine*::all(), 'stations' => *Station*::all()]);  }  } |
| --- |

Trong file code trên:

* Hàm index(): Trả về file index (file giao diện chính)

#### **AuthController**

File: **app/Http/Controllers/AuthController.php**

| class AuthController extends *Controller*  {  public function login()  {  return view('login');  }  public function checklogin(*LoginRequest* $request)  {  $user = *User*::where('Username', $request->username)->first();  if ($user && *Hash*::check($request->password, $user->Password)) {  *Auth*::login($user);  $name = *Auth*::user()->Name;  $request->session()->put('name', $name);  return redirect()->intended('')->with('type', 'success')  ->with('message', 'Đăng nhập thành công');  } else {  return redirect()->back()->with('type', 'warning')->with('message', 'Email hoặc mật khẩu không chính xác');  }  }  public function logout()  {  *Auth*::logout();  session()->flush();  return redirect()->route('login')->with('type', 'success')->with('message', 'Đăng xuất thành công');  }  } |
| --- |

* Hàm login(): Sử dụng phương thức Attempt của thư viện Auth để kiểm tra, phương thức này sẽ kiểm tra xem thông tin đăng nhập có khớp với một người dùng hiện tại trong cơ sở dữ liệu hay không. Nếu thông tin đúng, người dùng sẽ được xác thực và tạo một biến $name được lưu vào session store, sau đó điều hướng về trang chủ với thông báo ‘Đăng nhập thành công’
* Hàm logout(): Sử dụng phương thức "logout" của lớp "Auth" để đăng xuất người dùng hiện tại. Phương thức này sẽ xóa thông tin xác thực và session liên quan đến người dùng.

#### **RawMaterialController**

File: **app\Http\Controllers\RawMaterialController.php**

| use *App\Http\Requests\Material\MaterialStoreRequest*;  use *App\Http\Requests\Material\MaterialUpdateRequest*;  use *App\Models\RawMaterial*;  use *App\Models\RawMaterialType*;  use *Illuminate\Http\Request*;  use *Illuminate\Support\Facades\DB*;  use *Illuminate\Support\Facades\Session*;  class RawMaterialController extends *Controller*  {  public function index()  {  if (!*Session*::has("type") && !*Session*::has("message")) {  *Session*::flash('type', 'info');  *Session*::flash('message', 'Quản lý nguyên liệu thô');  }  $types = *RawMaterialType*::all();  return view('rawMaterials.index', compact('types'));  }  public function create()  {  $data = *RawMaterialType*::all();  return view('rawMaterials.create', compact('data'));  }  public function store(*MaterialStoreRequest* $request)  {  $validator = $request->validated();  $lastID = *DB*::table('RawMaterial')->max('Id\_RawMaterial');  if ($lastID === null) {  $id = 0; *// Gán giá trị mặc định cho biến $id nếu kết quả là NULL*  } else {  $id = $lastID + 1;  }  *DB*::table('RawMaterial')->insert([  'Id\_RawMaterial' => $id,  'Name\_RawMaterial' => $validator['Name\_RawMaterial'],  'Unit' => $validator['Unit'],  'Count' => $validator['Count'],  'FK\_Id\_RawMaterialType' => $validator['FK\_Id\_RawMaterialType'],  ]);  return redirect()->route('rawMaterials.index')->with([  'type' => 'success',  'message' => 'Nguyên liệu thô được tạo thành công'  ]);  }  public function show(string $id)  {  $material = *RawMaterial*::where('Id\_RawMaterial', $id)->first();  return view('rawMaterials.show', compact('material'));  }  public function edit(string $id)  {  $material = *RawMaterial*::where('Id\_RawMaterial', $id)->first();  $data = *RawMaterialType*::all();  return view('rawMaterials.edit', compact('material', 'data'));  }  public function update(*MaterialUpdateRequest* $request, string $id)  {  $validator = $request->validated();  *RawMaterial*::where('Id\_RawMaterial', '=', $id)->update([  'Name\_RawMaterial' => $validator['Name\_RawMaterial'],  'Unit' => $validator['Unit'],  'count' => $validator['count'],  'FK\_Id\_RawMaterialType' => $validator['FK\_Id\_RawMaterialType'],  ]);  return redirect()->route('rawMaterials.index')->with([  'message' => 'Sửa nguyên liệu thô thành công',  'type' => 'success'  ]);  }  public function destroy(string $id)  {  *RawMaterial*::where('Id\_RawMaterial', '=', $id)->delete();  return redirect()->route('rawMaterials.index')->with([  'message' => 'Xóa nguyên liệu thô thành công',  'type' => 'danger',  ]);  }  public function showRawMaterialsByType(*Request* $request)  {  if ($request->ajax()) {  $data = *DB*::table('RawMaterialType')  ->select('Id\_RawMaterial', 'Name\_RawMaterial', 'Unit', 'count', 'Name\_RawMaterialType')  ->join('RawMaterial', 'FK\_Id\_RawMaterialType', '=', 'Id\_RawMaterialType')  ->where('Id\_RawMaterialType', '=', $request->id)  ->get();  return response()->json([  'status' => 'success',  'data' => $data,  ]);  }  }  public function showUnit(*Request* $request)  {  if ($request->ajax()) {  $unit = *DB*::table('RawMaterial')  ->where('Id\_RawMaterial', $request->input('id'))  ->value('RawMaterial.Unit');  return response()->json([  'status' => 'success',  'unit' => $unit,  ]);  }  }  } |
| --- |

Trong file code trên:

* Hàm index(): Tạo biến $types để hiển thị ra “Loại nguyên vật liệu” cho người dùng chọn để hiển thị
* Hàm create(): Tạo biến $data để hiển thị ra “Loại nguyên vật liệu” để người dùng có thể chọn loại nguyên vật liệu trong form “Thêm nguyên vật liệu thô”.
* Hàm store(): Tạo một biến $material mới và gán dữ liệu cho nó là những trường người dùng nhập trong form, dữ liệu được trả về biến $request của class MaterialStoreRequest (tự tạo) sau đó lưu lại, cuối cùng là điều hướng về trang chủ với thông báo “Nguyên liệu thô được tạo thành công”
* Hàm show(): Tìm nguyên liệu thô mà người dùng muốn xem chi tiết bằng câu truy vấn (Sử dụng Eloquent) và sau đó trả về form hiển thị chi tiết với biến $material vừa tạo.
* Hàm edit(): Tìm nguyên liệu thô mà người dùng muốn sửa bằng câu truy vấn (Sử dụng Eloquent) và tạo biến $data để hiển thị ra “Loại nguyên vật liệu” để người dùng có thể sửa loại nguyên vật liệu trong form “Sửa nguyên vật liệu thô”.
* Hàm update(): Tạo một biến $validator là những dữ liệu mà người dùng điền trong form sửa (từ biến $request của class MaterialUpdateRequest) sau đó sử dụng câu truy vấn (Sử dụng Eloquent) để cập nhật dữ liệu, sau đó là điều hướng về trang chủ với thông báo “Sửa nguyên liệu thô thành công”.
* Hàm destroy(): Tìm nguyên liệu thô mà người dùng muốn xóa bằng câu truy vấn (Sử dụng Eloquent) và xóa đi, sau đó là điều hướng về trang chủ với thông báo “Xóa nguyên liệu thô thành công”
* Hàm showRawMaterialsByType(): Lấy Tên nguyên liệu, đơn vị, số lượng và tên loại nguyên liệu rồi trả dữ liệu về giao diện theo kiểuJSON.
* Hàm showUnit(): Lấy đơn vị sau đó trả dữ liệu về giao diện theo kiểu JSON

#### **OrderController**

File: **app\Http\Controllers\OrderController.php**

| use *App\Models\Order*;  use *App\Models\Customer*;  use *App\Models\RawMaterial*;  use *App\Models\ContainerType*;  use *Illuminate\Http\Request*;  use *Illuminate\Support\Facades\Session*;  use *Illuminate\Support\Facades\DB*;  class OrderController extends *Controller*  {  public function index()  {  if (!*Session*::has("type") && !*Session*::has("message")) {  *Session*::flash('type', 'info');  *Session*::flash('message', 'Quản lý đơn thùng hàng');  }  $data = *Order*::where('isSimple', 1)->paginate(5);  return view('simples.index', compact('data'));  }  public function show(*Order* $order)  {  $data = *DB*::table('ContentSimple')  ->select(  'RawMaterial.Name\_RawMaterial',  'ContentSimple.Count\_RawMaterial',  'RawMaterial.unit',  'ContainerType.Name\_ContainerType',  'ContentSimple.Count\_Container',  'ContentSimple.Price\_Container'  )  ->join('RawMaterial', 'ContentSimple.FK\_Id\_RawMaterial', '=', 'RawMaterial.Id\_RawMaterial')  ->join('ContainerType', 'ContentSimple.FK\_Id\_ContainerType', '=', 'ContainerType.Id\_ContainerType')  ->join('Order', 'ContentSimple.FK\_Id\_Order', '=', 'Order.Id\_Order')  ->where('FK\_Id\_Order', $order->Id\_Order)  ->get();  return view('simples.show', compact('order', 'data'));  }  public function create()  {  $customers = *Customer*::get();  $containers = *ContainerType*::get();  $materials = *RawMaterial*::get();  return view('simples.create', [  'customers' => $customers,  'containers' => $containers,  'materials' => $materials,  ]);  }  public function edit(*Order* $order)  {  $customers = *Customer*::get();  $containers = *ContainerType*::get();  $materials = *RawMaterial*::get();  $data = *DB*::table('ContentSimple')  ->select(  'ContentSimple.Id\_SimpleContent',  'RawMaterial.Name\_RawMaterial',  'ContentSimple.FK\_Id\_RawMaterial',  'ContentSimple.Count\_RawMaterial',  'RawMaterial.unit',  'ContentSimple.FK\_Id\_ContainerType',  'ContainerType.Name\_ContainerType',  'ContentSimple.Count\_Container',  'ContentSimple.Price\_Container'  )->join('RawMaterial', 'ContentSimple.FK\_Id\_RawMaterial', '=', 'RawMaterial.Id\_RawMaterial')  ->join('ContainerType', 'ContentSimple.FK\_Id\_ContainerType', '=', 'ContainerType.Id\_ContainerType')  ->join('Order', 'ContentSimple.FK\_Id\_Order', '=', 'Order.Id\_Order')  ->where('FK\_Id\_Order', $order->Id\_Order)  ->get();  return view('simples.edit', [  'order' => $order,  'customers' => $customers,  'containers' => $containers,  'materials' => $materials,  'data' => $data,  ]);  }  public function destroy(*Order* $order)  {  *DB*::table('ContentSimple')->where('FK\_Id\_Order', $order->Id\_Order)->delete();  $order->delete();  return redirect()->route('orders.index')->with([  'message' => 'Xóa đơn hàng thành công',  'type' => 'danger',  ]);  }  public function storeOrder(*Request* $request)  {  if ($request->ajax()) {  $formData = $request->input('formData');  $data = [];  parse\_str($formData, $formDataArray);  $lastOrderId = *DB*::table('Order')->max('Id\_Order');  if ($lastOrderId === null) {  $id = 1; *// Gán giá trị mặc định cho biến $id nếu kết quả là NULL*  } else {  $id = $lastOrderId + 1;  }  $formDataArray['Id\_Order'] = $id;  *DB*::table('order')->insert([  'Id\_Order' => $id,  'FK\_Id\_Customer' => $formDataArray['FK\_Id\_Customer'],  'Date\_Order' => $formDataArray['Date\_Order'],  'Date\_Dilivery' => $formDataArray['Date\_Dilivery'],  'Date\_Reception' => $formDataArray['Date\_Reception'],  'Note' => $formDataArray['Note'],  'IsSimple' => $formDataArray['isSimple'],  ]);  return response()->json([  'status' => 'success',  'id' => $id,  'data' => $formDataArray  ]);  }  }  public function updateOrder(*Request* $request)  {  if ($request->ajax()) {  $formData = $request->input('formData');  $id = $request->input('id');  $data = [];  parse\_str($formData, $formDataArray);  $data[] = $formDataArray;  *DB*::table('Order')->where('Id\_Order', $id)->update([  'FK\_Id\_Customer' => $formDataArray['FK\_Id\_Customer'],  'Date\_Order' => $formDataArray['Date\_Order'],  'Date\_Dilivery' => $formDataArray['Date\_Dilivery'],  'Date\_Reception' => $formDataArray['Date\_Reception'],  'Note' => $formDataArray['Note'],  ]);  return response()->json([  'status' => 'success',  'id' => $id,  ]);  }  }  } |
| --- |

Trong file code trên:

* Hàm index(): Tạo một biến $data (Lấy những dữ liệu có trường ‘IsSimple’ = 1, tức là những đơn hàng chỉ có thùng hàng) từ bảng ‘Orders’ rồi trả về giao diện quản lý
* Hàm show(): Lấy dữ liệu tương ứng với cả biến $order được trả về từ file quản lý xong sử dụng Query Builder của Laravel để lấy ra dữ liệu tương ứng sau đó điều hướng đến giao diện hiển thị chi tiết.
* Hàm create(): Tạo 3 biến $customers, $containers, $materials (3 biến này để hiển thị Khách hàng, Thùng chứa và Nguyên liệu cho người dùng chọn) rồi trả về giao diện thêm đơn hàng.
* Hàm edit(): Tạo 3 biến $customers, $containers, $materials (3 biến này để hiển thị Khách hàng, Thùng chứa và Nguyên liệu cho người dùng chọn), sau đó sử dụng Query Builder để lấy ra những dữ liệu cần thiết để hiển thị rồi trả tất cả về form sửa đơn hàng.
* Hàm destroy(): Sử dụng Query Builder để xóa đơn hàng theo Id\_Order được trả về từ file quản lý, sau đó điều hướng lại về file quản lý với thông báo ‘Xóa đơn hàng thành công’.
* Hàm storeOrder(): Tạo biến $id là ID cao nhất trong bảng ‘Order’ bằng QueryBuilder, rồi tăng 1 đơn vị của biến $id, sau đó thêm dữ liệu vào database với những dữ liệu người dùng nhập trong form tạo đơn hàng.
* Hàm updateOrder(): Cập nhật dữ liệu bằng Query Builder (dữ liệu được gửi vào bằng Ajax).

#### **ContentSimpleController**

File: **app\Http\Controllers\ContentSimpleController.php**

| use *Illuminate\Http\Request*;  use *Illuminate\Support\Facades\DB*;  class ContentSimpleController extends *Controller*  {  public function addSimple(*Request* $request)  {  if ($request->ajax()) {  $formData = $request->input('formData');  $unit = $request->input('unit');  $data = [];  parse\_str($formData, $formDataArray);  $formattedPrice = number\_format($formDataArray['Price\_Container'], 0, ',', '.') . ' VNĐ';  $formDataArray['unit'] = $unit;  $formDataArray['formattedPrice'] = $formattedPrice;  $data[] = $formDataArray;  return response()->json([  'status' => 'success',  'data' => $data,  ]);  }  }  public function storeSimple(*Request* $request)  {  if ($request->ajax()) {  $rowData = $request->input('rowData');  foreach ($rowData as $row) {  $lastOrderId = *DB*::table('ContentSimple')->max('Id\_SimpleContent');  if ($lastOrderId === null) {  $id = 1; *// Gán giá trị mặc định cho biến $id nếu kết quả là NULL*  } else {  $id = $lastOrderId + 1;  }  *DB*::table('ContentSimple')->insert([  'Id\_SimpleContent' => $id,  'FK\_Id\_RawMaterial' => $row['FK\_Id\_RawMaterial'],  'Count\_RawMaterial' => $row['Count\_RawMaterial'],  'FK\_Id\_ContainerType' => $row['FK\_Id\_ContainerType'],  'Count\_Container' => $row['Count\_Container'],  'Price\_Container' => $row['Price\_Container'],  'FK\_Id\_Order' => $row['FK\_Id\_Order'],  'ContainerProvided' => 0,  'PedestalProvided' => 0,  'RFIDProvided' => 0,  'RawMaterialProvided' => 0,  ]);  }  return response()->json([  'status' => 'success',  ]);  }  }  public function updateSimple(*Request* $request)  {  if ($request->ajax()) {  $rowData = $request->input('rowData');  foreach ($rowData as $row) {  $id = $row['Id\_SimpleContent'];  *DB*::table('ContentSimple')->where('Id\_SimpleContent', $id)->update([  'FK\_Id\_RawMaterial' => $row['FK\_Id\_RawMaterial'],  'Count\_RawMaterial' => $row['Count\_RawMaterial'],  'FK\_Id\_ContainerType' => $row['FK\_Id\_ContainerType'],  'Count\_Container' => $row['Count\_Container'],  'Price\_Container' => $row['Price\_Container'],  ]);  }  return response()->json([  'status' => 'success',  ]);  }  }  public function deleteSimple(*Request* $request)  {  if ($request->ajax()) {  $id = $request->input('id');  *DB*::table('ContentSimple')->where('Id\_SimpleContent', $id)->delete();  return response()->json([  'status' => 'success'  ]);  }  }  } |
| --- |

Trong file code trên:

* Hàm addSimple(): Tạo một biến $data chứa những dữ liệu người dùng nhập trong form “Thông tin thùng chứa” (được trả vào bằng Ajax) sau đó thêm trường ‘unit’ và trường ‘formattedPrice’ rồi trả lại dữ liệu về giao diện rồi hiển thị tại bảng.
* Hàm storeSimple(): Tạo biến $id là ID cao nhất trong bảng ‘ContentSimple’ bằng QueryBuilder, rồi tăng 1 đơn vị của biến $id, sau đó thêm dữ liệu vào database với những dữ liệu người dùng nhập trong form “Thông tin thùng chứa” (được trả vào bằng Ajax).
* Hàm updateSimple(): Lấy dữ liệu từ giao diện trả về, sau đó chạy vòng lặp để cập nhật dữ liệu vào máy chủ.
* Hàm deletSimple(): Lấy $id được trả về từ giao diện (Trả về bằng Ajax) sau đó xóa luôn Thùng hàng có $id đó bằng Query Builder.

#### **ContentPackController**

File: **app\Http\Controllers\ContentPackController.php**

| use *App\Models\Order*;  use *App\Models\Customer*;  use *App\Models\ContainerType*;  use *App\Models\RawMaterial*;  use *Illuminate\Http\Request*;  use *Illuminate\Support\Facades\Session*;  use *Illuminate\Support\Facades\DB*;  class ContentPackController extends *Controller*  {  public function index()  {  if (!*Session*::has("type") && !*Session*::has("message")) {  *Session*::flash('type', 'info');  *Session*::flash('message', 'Quản lý đơn gói hàng');  }  $data = *Order*::where('isSimple', 0)->paginate(5);  return view('packs.index', compact('data'));  }  public function create()  {  $customers = *Customer*::get();  if (isset($\_GET['id'])) {  $id = $\_GET['id'];  $information = *DB*::table('Order')  ->select('FK\_Id\_Customer', 'Date\_Order', 'Date\_Dilivery', 'Date\_Reception', 'Note')  ->where('Id\_Order', $id)  ->first();  $data = *DB*::table('ContentPack')->select('Id\_PackContent', 'Count\_Pack', 'Price\_Pack')  ->where('FK\_Id\_Order', $id)  ->get();  return view('packs.create', [  'customers' => $customers,  'information' => $information,  'data' => $data,  'count' => 1,  ]);  } else {  return view('packs.create', [  'customers' => $customers,  ]);  }  }  public function createPack()  {  $containers = *ContainerType*::get();  $materials = *RawMaterial*::get();  return view('packs.createPack', [  'containers' => $containers,  'materials' => $materials,  ]);  }  public function storePack(*Request* $request)  {  if ($request->ajax()) {  $data = $request->input('packData');  $lastOrderId = *DB*::table('ContentPack')->max('Id\_PackContent');  if ($lastOrderId === null) {  $id = 1; *// Gán giá trị mặc định cho biến $id nếu kết quả là NULL*  } else {  $id = $lastOrderId + 1;  }  *DB*::table('ContentPack')->insert([  'Id\_PackContent' => $id,  'Count\_Pack' => $data['Count\_Pack'],  'Price\_Pack' => $data['Price\_Pack'],  'FK\_Id\_Order' => $data['FK\_Id\_Order'],  'HaveEilmPE' => 0,  'HaveNFC' => 0,  ]);  return response()->json([  'status' => 'success',  'id' => $id,  ]);  }  }  public function storeDetailContentSimpleOfPack(*Request* $request)  {  if ($request->ajax()) {  $OrderID = $request->input('FK\_Id\_Order');  $packID = $request->input('Id\_PackContent');  $IdArr = *DB*::table('ContentSimple')  ->where('FK\_Id\_Order', $OrderID)  ->pluck('Id\_SimpleContent')  ->toArray();  foreach ($IdArr as $each) {  *DB*::table('DetailContentSimpleOfPack')->insert([  'FK\_Id\_SimpleContent' => $each,  'FK\_Id\_PackContent' => $packID,  ]);  }  return response()->json([  'status' => 'success',  'id' => $OrderID,  ]);  }  }  public function deletePack(*Request* $request)  {  if ($request->ajax()) {  $id = $request->input('id');  $Id\_Order = $request->input('Id\_Order');  *DB*::table('DetailContentSimpleOfPack')->where('FK\_Id\_PackContent', $id)->delete();  *DB*::table('ContentSimple')->where('FK\_Id\_Order', $Id\_Order)->delete();  *DB*::table('ContentPack')->where('FK\_Id\_Order', $Id\_Order)->delete();  return response()->json([  'status' => 'success'  ]);  }  }  } |
| --- |

Trong file code trên:

* Hàm index(): Tạo một biến $data (Lấy những dữ liệu có trường ‘IsSimple’ 0, tức là những đơn hàng là gói hàng) từ bảng ‘Orders’ rồi trả về giao diện quản lý
* Hàm create(): Tạo biến customers cho người dùng có thể chọn dữ liệu tương ứng trong giao diện. Sẽ có 2 trường hợp, một là trường hợp người dùng trở về từ form “Thêm thùng hàng vào gói hàng”, trường hợp hai là người dùng mới đến từ form “Quản lý đơn gói hàng”, thì tại trường hợp 1 sẽ hiển thị thông tin tương ứng với giao diện, còn trường hợp 2 thì sẽ để tất cả ô input trống.
* Hàm createPack(): Trả về form “Thêm thùng hàng vào gói hàng” với biến, containers và materials cho người dùng có thể chọn dữ liệu tương ứng trong giao diện.
* Hàm storePack(): Lấy dữ liệu bằng Ajax và thêm vào cơ sở dữ liệu vào bảng ContentPack
* Hàm storeDetailContentSimpleOfPack(): Lấy dữ liệu bằng Ajax và thêm dữ liệu vào bảng ‘DetailContentSimpleOfPack’.
* Hàm deletePack(): Xóa dữ liệu tại các bảng: DetailContentSimpleOfPack, ContentSimple và ContentPack tương ứng với gói hàng mà người dùng muốn xóa.

#### **OrderMakeOrExpeditionController**

File: **app\Http\Controllers\OrderMakeOrExpeditionController .php**

| use *Illuminate\Http\Request*;  use *Illuminate\Support\Carbon*;  use *Illuminate\Support\Facades\DB*;  use *Illuminate\Support\Facades\Session*;  use *App\Models\OrderMakeOrExpedition*;  class OrderMakeOrExpeditionController extends *Controller*  {  public function index()  {  if (!*Session*::has("type") && !*Session*::has("message")) {  *Session*::flash('type', 'info');  *Session*::flash('message', 'Quản lý đơn sản xuất và giao hàng');  }  $data = *OrderMakeOrExpedition*::paginate(5);  return view('makesOrExpeditions.index', compact('data'));  }  public function create()  {  $data = *DB*::table('Order')  ->join('ContentSimple', 'Order.Id\_Order', '=', 'ContentSimple.FK\_Id\_Order')  ->join('Customer', 'Order.FK\_Id\_Customer', '=', 'Customer.Id\_Customer')  ->join('ContainerType', 'ContentSimple.FK\_Id\_ContainerType', '=', 'ContainerType.Id\_ContainerType')  ->where('ContentSimple.ContainerProvided', 0)  ->where('ContentSimple.PedestalProvided', 0)  ->where('Order.isSimple', 1)  ->whereNotIn('ContentSimple.Id\_SimpleContent', function ($query) {  $query->select('FK\_Id\_ContentSimple')  ->from('ProcessContentSimple');  })  ->select('Order.Id\_Order', 'Id\_SimpleContent', 'Customer.Name\_Customer', 'ContainerType.Name\_ContainerType', 'ContentSimple.Count\_Container', 'ContentSimple.Price\_Container')  ->get();  $orders = *OrderMakeOrExpedition*::orderBy('Id\_OrderMakeOrExpedition', 'desc')  ->where('SimpleOrPack', 0)  ->limit(200)  ->paginate(10);  return view('makesOrExpeditions.simples.create', compact('data', 'orders'));  }  public function store(*Request* $request)  {  if ($request->ajax()) {  $rowData = $request->input('rowData');  $dateDilivery = $request->input('dateValue');  $lastOrderId = *DB*::table('OrderMakeOrExpedition')->max('Id\_OrderMakeOrExpedition');  if ($lastOrderId === null) {  $id = 1; *// Gán giá trị mặc định cho biến $id nếu kết quả là NULL*  } else {  $id = $lastOrderId + 1;  }  $data = *OrderMakeOrExpedition*::create([  'Id\_OrderMakeOrExpedition' => $id,  'Count' => 1,  'DateDilivery' => $dateDilivery,  'SimpleOrPack' => 0,  'MakeOrExpedition' => 0,  'Data\_Start' => *Carbon*::now(),  ]);  if ($data) {  foreach ($rowData as $row) {  *DB*::table('DetailContentSimpleOrderMakeOrExpedition')->insert([  'FK\_Id\_ContentSimple' => $row['Id\_SimpleContent'],  'FK\_Id\_OrderMakeOrExpedition' => $id,  ]);  }  return response()->json([  'status' => 'success',  'data' => $data,  'id' => $id,  ]);  }  }  }  public function show(*Request* $request)  {  if ($request->ajax()) {  $id = $request->input('id');  $data = *DB*::table('ContentSimple')  ->select('Name\_RawMaterial', 'Count\_RawMaterial', 'Name\_ContainerType', 'Count\_Container', 'Price\_Container')  ->join('RawMaterial', 'Id\_RawMaterial', '=', 'FK\_Id\_RawMaterial')  ->join('ContainerType', 'Id\_ContainerType', '=', 'FK\_Id\_ContainerType')  ->join('DetailContentSimpleOrderMakeOrExpedition', 'Id\_SimpleContent', '=', 'FK\_Id\_ContentSimple')  ->where('FK\_Id\_OrderMakeOrExpedition', '=', $id)  ->get();  return response()->json([  'status' => 'success',  'data' => $data,  ]);  }  }  public function destroy(*Request* $request)  {  if ($request->ajax()) {  $rowData = $request->input('rowData');  foreach ($rowData as $row) {  *DB*::table('DetailContentSimpleOrderMakeOrExpedition')  ->where('FK\_Id\_OrderMakeOrExpedition', '=', $row['Id\_OrderMakeOrExpedition'])  ->delete();  *DB*::table('OrderMakeOrExpedition')  ->where('Id\_OrderMakeOrExpedition', '=', $row['Id\_OrderMakeOrExpedition'])  ->delete();  }  return response()->json([  'status' => 'success',  ]);  }  }  } |
| --- |

Trong file code trên:

* Hàm index(): Tạo biến $data lấy dữ liệu từ bảng “OrderMakeOrExpedition” và trả về giao diện.
* Hàm create(): Tạo 2 biến tương ứng để người dùng có thể sử dụng và trả về giao diện tạo đơn sản xuất
* Hàm store(): Lưu những dữ liệu mà người dùng chọn (Được gửi qua từ Ajax) và lưu vào máy chủ
* Hàm show(): Lấy ID của đơn sản xuất và lấy ra dữ liệu (Được gửi qua từ Ajax) tương ứng xong trả về giao diện để hiển thị chi tiết.
* Hàm destroy(): Lấy ID (Được gửi qua từ Ajax) tương ứng xong xóa dữ liệu của 2 bảng OrderMakeOrExpedition và DetailContentSimpleOrderMakeOrExpedition.

## **Bước 8:** Viết các hàm trong file Request

* File StoreRequest (Để validation form đăng nhập)

Php artisan make:request LoginRequest

File: **app\Http\Requests\LoginRequest.php**

class LoginRequest extends *FormRequest*

{

public function authorize(): bool

{

return true;

}

*/\*\**

*\* Get the validation rules that apply to the request.*

*\**

*\* @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array<mixed>|string>*

*\*/*

public function rules(): array

{

return [

'username' => [

'required'

],

'password' => [

'required'

],

];

}

public function messages()

{

return [

'username.required' => 'Vui lòng nhập username',

'password.required' => 'Vui lòng nhập password'

];

}}

Trong file code trên:

* Hàm authorize(): Để xác nhận quyền của người dùng.
* Hàm rules(): Trả về một mảng chứa các quy tắc để validation.
  + Trường ‘username’, ‘password’ sẽ bắt buộc người dùng phải nhập.
* Hàm messages(): Trả về một mảng chứa các thông báo.
  + Trường 'username.required’: 'Vui lòng nhập username'
  + Trường 'password.required': 'Vui lòng nhập password'
* File MaterialStoreRequest

Php artisan make:request Material.MaterialStoreRequest

File: **app\Http\Requests\Material\MaterialStoreRequest.php**

class MaterialStoreRequest extends *FormRequest*

{

*/\*\**

*\* Determine if the user is authorized to make this request.*

*\*/*

public function authorize(): bool

{

return true;

}

*/\*\**

*\* Get the validation rules that apply to the request.*

*\**

*\* @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array<mixed>|string>*

*\*/*

public function rules(): array

{

return [

'Name\_RawMaterial' => [

'required',

],

'Unit' => [

'required',

],

'count' => [

'required',

'numeric',

],

'FK\_Id\_RawMaterialType' => [

'required',

function ($attribute, $value, $message) {

if ($value === 'default') {

$message('Vui lòng chọn loại nguyên liẹu.');

}

},

],

];

}

public function messages(): array

{

return [

'Name\_RawMaterial.required' => 'Vui lòng nhập tên nguyên liệu thô',

'Unit.required' => 'Vui lòng nhập đơn vị',

'count.required' => 'Vui lòng nhập số lượng',

'FK\_Id\_RawMaterialType.required' => 'Vui lòng chọn loại nguyên liệu',

];

}

}

Trong file code trên:

* Hàm authorize(): Để xác nhận quyền của người dùng.
* Hàm rules(): Trả về một mảng chứa các quy tắc để validation.
  + Trường ‘Name\_RawMaterial’, ‘Unit’, sẽ bắt buộc người dùng phải nhập.
  + Trường ‘count’: Sẽ bắt buộc người dùng phải nhập, và dữ liệu nhập vào phải là số.
  + Trường ‘FK\_Id\_RawMaterialType’: Sẽ bắt buộc người dùng phải chọn select loại nguyên liệu, không được để mặc định.
* Hàm messages(): Trả về một mảng chứa các thông báo.
  + Trường ‘Name\_RawMaterial.required’: Nếu người dùng không nhập tên nguyên vật liệu thì sẽ thông báo “Vui lòng nhập tên nguyên liệu thô”
  + Trường Unit.required’: Nếu người dùng không nhập đơn vị thì sẽ thông báo “Vui lòng nhập đơn vị”
  + Trường “count.required”: Nếu người dùng không nhập số lượng thì sẽ thông báo “Vui lòng nhập số lượng”
  + Trường “FK\_Id\_RawMaterialType.required”: Nếu người không chọn thi fsex thông báo “Vui lòng chọn loại nguyên liệu”.

# **Thực hiện chương trình**

* Mở cmd lên tại thư mục chính.

php artisan serve

* Mở trình duyệt lên tìm trên thanh địa chỉ: localhost:8000